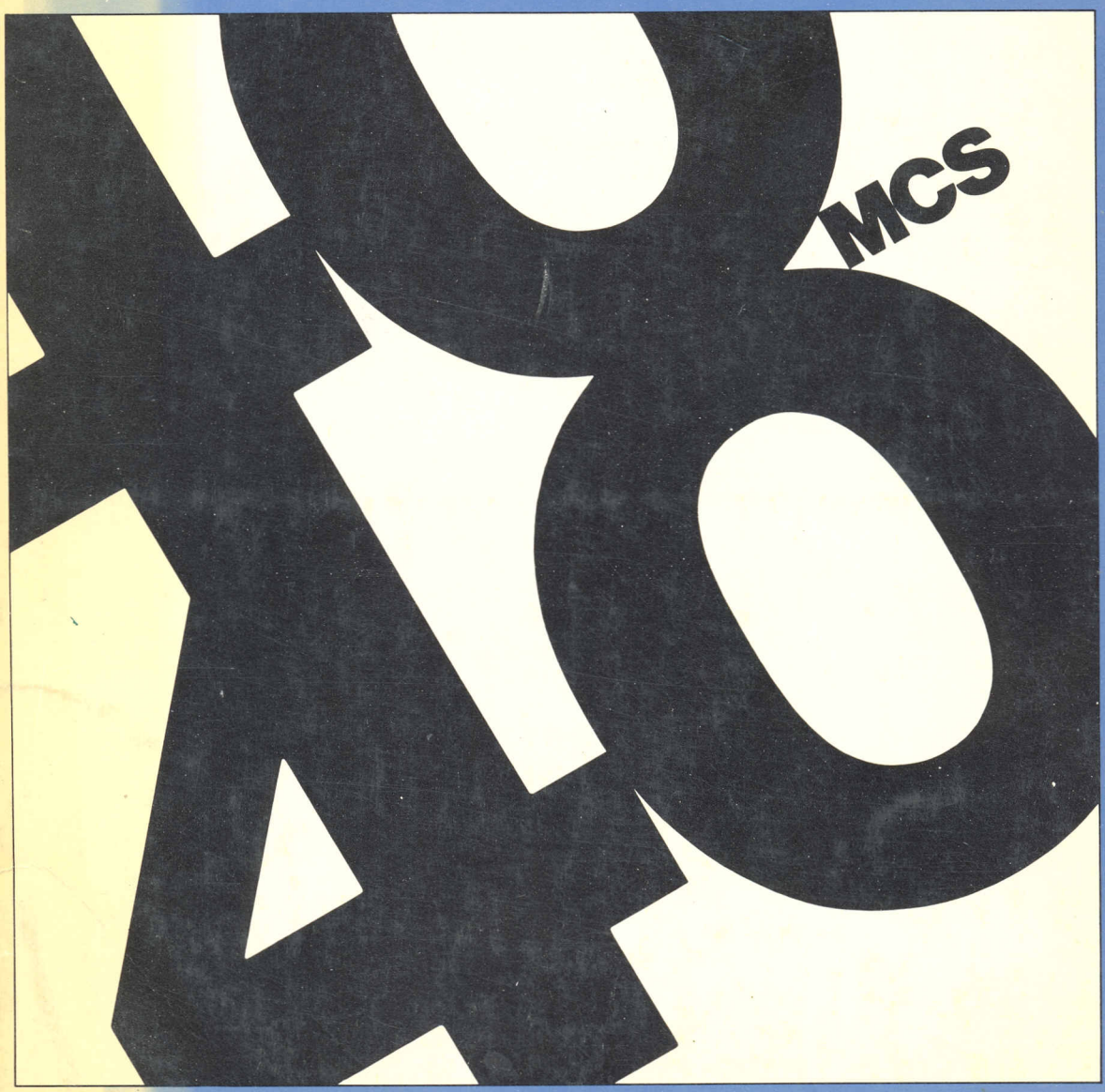
intel® MCS-40™ User's Manual

Third Edition
March 1976



98-042C

intel®
MCS-40™
User's Manual

The Intel MCS-40 family offers a track record beginning in 1971 of deliverability, reliability and complete support of both large and small customers. Now Intel has made two major additions to the MCS-40 Microcomputer component family with the introduction of the 4269 Programmable Keyboad/Display device and the 4265 Programmable General Purpose I/O device.

The 4269 Programmable Keyboard/Display device provides extensive support of keyboard and display devices resulting in significant savings in program space, program performance and design development time. The keyboard and display support feature includes automatic key scan, 2-key rollover, key debounce and automatic display refresh.

The 4265 GP I/O device can be configured into any one of fourteen different I/O devices by a single software instruction. The 4265 has individual bit set/reset capability and can be configured to interface 8-bit peripheral or I/O chips, 8-bit microprocessors such as the 8080, and standard RAM memories. It can also be configured as a synchronous, asynchronous or unbuffered general I/O interface.

This third edition of the MCS-40 User's Manual in itself represents a major extension of the total product and product support package offer by Intel to MCS-40 microcomputer users. This new edition contains complete, separate data sheets for all MCS-40 components, including those interfaceable to 4004/4040 via the 4265 GP I/O (e.g., the 8251 Programmable Serial Communications device, the 8253 Programmable Timer, etc.). Besides extensions to all chapters of the manual, there is also a new applications chapter containing applications notes on the 4265, 4269 and 5101 low-power memory.

Intel will appreciate your comments on this new MCS-40 User's Manual and any other aspect of the MCS-40 microcomputer family.

intel® Microcomputers. First from the beginning.

CONTENTS

*MCS-40 is an Intel trademark.

INTRODUCTION

Since their inception, digital computers have continuously become more efficient, expanding into new applications with each major technological improvement. The advent of minicomputers enabled the inclusion of digital computers as a permanent part of various process control systems. Unfortunately, the size and cost of minicomputers in "dedicated" applications has limited their use. Another approach has been the use of custom built systems made up of "random logic" (i.e., logic gates, flip-flops, counters, etc.). However, the huge expense and development time involved in the design and debugging of these systems has restricted their use to large volume applications where the development costs could be spread over a large number of machines.

Today, Intel offers the systems designer a new alternative . . . . the microcomputer. Utilizing the technologies and experience gained in becoming the world's largest supplier of LSI memory components, Intel has made the power of the digital computer available at the integrated circuit level.

ECONOMICS OF USING MICROCOMPUTERS

The MCS-40™ microcomputers built around the 4004 and 4040 CPUs are a new extension of computer technology which offer users exciting possibilities for creating new products and services. Engineers are becoming more aware of the ways in which microcomputers can be applied to solve their problems. There are five basic reasons why many engineers have begun to use the MCS-40. These are:

1. Manufacturing costs of products can be significantly reduced.
2. Products can get to the market faster providing a company with the opportunity to increase product sales and market share.
3. Product capability is enhanced allowing manufacturers to provide customers with better products which can frequently command a higher price in the market place.
4. Development costs and time are reduced.
5. Product reliability is increased which leads to a corresponding reduction in both service and warranty costs.

Microcomputers simplify almost every phase of product development. The first step, as in any product design program, is to identify the various functions that the end system is expected to perform. These functions are then implemented by encoding suitable sequences of instructions (programs) in the memory elements. Data and certain types of programs will be stored in RAM circuits, while the basic program will be stored in ROM circuits. The microprocessor performs all of the system's functions by fetching the instructions in memory, executing them and communicating the results via the microcomputer's I/O ports. A single-chip microprocessor, executing the programmed logic stored in a single ROM element, can perform the same logical functions that have previously required many logic gates.

How Memory Replaces Random Logic

The MCS-40™ microcomputer system replaces logic by storing program sequences in memory rather than implementing these sequences with gates and flip-flops. It can be said that 8 to 16 bits of memory are the logical equivalent of a single gate. Assuming that the type IC used today contains on the order of 10 gates, then one can conclude that logic can be stored in memory in a very cost effective fashion. The following table indicates the number of IC's which are replaced by a single ROM (Read Only Memory). The table was derived by using the assumptions that 8 to 16 bits of ROM replace a gate and that on the average an IC contains 10 gates.

ROM Memory Size Bits	Gates Replaced	IC's Replaced
2048	128-256	13-25
4096	256-512	25-50
8192	512-1024	50-100
16384	1024-2048	100-200

Table I. Number of IC's Replaced with a ROM (Read Only Memory)

REDUCING MANUFACTURING COSTS

If the burdened manufacturing cost of a digital electronic system is divided by the number of ICs, one generally finds that the system costs between $2 and $6 per IC to fabricate. The higher costs are generally associated with systems manufactured in volumes from 10 to 100 units annually. The table below presents a more detailed analysis of the source of these surprisingly high costs. The costs, themselves, are stated conservatively.

IC	.50
Incoming Inspection	.05
PC Card	.50
Fabrication	.05
Board Test and Rework	.10
Connector	.05
Discretes	.05
Wiring	.10
Power	.10
Cabinetry, Fans, Etc.	.10
	$1.60

Table II. System Manufacturing Costs Per IC

The ASP (average sale price) of an Integrated Circuit today is approximately 50¢. Incoming inspection and testing of these ICs costs the average company 5¢. However, many companies are now buying aged and tested circuits for their applications in order to increase system reliability. This adds about 15¢ to unit costs. Simple PC cards may cost as little as 25¢ an IC position, but the average cost in most applications for high quality cards is closer to 50¢. Sophisticated multilayer cards used in many high performance systems frequently cost *over a dollar a position.* When customers put ICs in sockets and then wire wrap cards, the cost per IC position quickly approaches $2. Customers with automatic IC insertion equipment and efficient flow soldering machines can fabricate a PC card for as low as 3¢ an IC position, though the average price is closer to 5¢. Board test and rework add another dime to system cost, while the cost of a connector divided by the number of ICs per printed circuit card frequently exceeds 5¢. In general, resistors, capacitors, power bus bars, etc., add a cost of 5¢ an IC position. Systems frequently average one wire or more per IC position and the wires put in with automatic equipment frequently cost over 10¢. Finally, the cost of power supplies and mechanical packaging add another 20¢ an IC position.

To determine the total savings in system manufacturing cost, the user must subtract the cost of implementing an equivalent system with a microcomputer. In moderate volumes, an MCS 40 with 16,384 bits of ROM, a processor, and a minimal amount of RAM can be purchased for under $40. This system has the potential of displacing between $150 and $600 of system manufacturing cost.

Reducing Development Time and Cost

The MCS-40™ system simplifies almost every phase of product development. Because of the extensive design aids and support supplied with microcomputers, it is relatively easy to develop application programs that tailor the device to the system. One of the most significant advantages of the MCS-40 system is modularity of the component family which aids in development. Development cycles can be cut by as long as six to twelve months. The table below tabulates a number of the steps in a development cycle and indicates how the MCS-40 system can affect them. Surprisingly, product definition is frequently speeded up once the decision has been made to use a microcomputer. This is because the incremental cost for adding features to the system is usually small and can be easily extimated. For example, added features such as automatic tax computation for an electronic cash register may only require the addition of a single ROM. The addition of one LSI chip has a minimal effect on total system cost, power and packaging requirements. On the other hand, the same function implemented with IC logic might require two or three fairly large PC cards filled with MSI and SSI.

System and logic design time is also reduced. Programming is a faster way to design than using logic diagrams. If you're used to logic design and the idea of designing with programmed logic seems like too radical a change, regardless of advantages, there's no need to worry because Intel has already done most of the groundwork for you. The INTELLEC® 4 Development Systems provide flexible, inexpensive and simplified methods for OEM product development. The INTELLEC® 4 system provides RAM program storage making program loading and modification easier, a display and control console for system monitoring and debugging, expandable memory and I/O, TTY interface, a PROM programming cap-

Development Steps	Conventional System	Programmed Logic
Product definition		Simplified because of ease of incorporating features
System and logic design	Done with logic diagrams	Can be programmed with design aids (compilers, assemblers, editors)
Debug	Done with conventional lab instrumentation	Software and hardware aids reduce time
PC card layout		Fewer cards to layout
Documentation		Less hardware to document
Cooling and packaging		Reduced system size and power consumption eases job
Power distribution		Less power to distribute
Engineering changes	Done with yellow wire	Change program in PROM

Table III. How Development Time and Cost are Reduced with Microcomputers

ability and a standard software package (System Monitor and Assembler). In addition to the standard software package available with the INTELLEC® 4 development system, Intel offers a cross-assembler and simulator written in FORTRAN IV and designed to run on any large scale computer. The programs may be procured directly from Intel or from a number of nation-wide computer time-sharing services. Intel's Microcomputer Systems Group and Regional Applications Engineers are available to provide assistance in every phase of your product development.

Intel provides complete documentation on all their hardware and software products. In addition to this User's Manual, there are the:

- 4004 and 4040 Assembly Language Programming Manual
- INTELLEC 4 Operator's Manual
- INTELLEC 4 Hardware and Microcomputer Modules Reference Manual

This User's Manual is intended as your primary source of information on the 4004 and 4040 processors and their family components.

The above aids also reduce the time for system debugging. PC card layout time is reduced simply because there are fewer cards to lay out. This reduction in hardware also reduces the load on the technical writers who must develop maintenance manuals. Parts lists become shorter, easing the task of transferring the product to manufacturing. Cooling, packaging, and power distribution problems frequently become trivial. Finally, engineering changes that are difficult to make and frequently tedious to document, become simple program changes. These can be made by changing the pattern in a ROM or PROM (Programmable Read Only Memory) such as Intel's 4702A.

Products Can Get to the Market Faster

The use of the MCS-40™ system allows the user to have a shorter development cycle, thus reducing development time and allows a faster response to the market from conception to production. This affords the MCS-40 user a definite advantage.

High Product Prices Due to Enhanced Product Capability

Product features can be easily added to microcomputer systems by simply adding more program storage. Many MCS-40 users have utilized this characteristic of microcomputers as a way of increasing the value of their product without significantly adding to the cost of the product. Examples of such easily added features are: putting automatic tax computations into a cash register by adding more ROM, adding automatic calibration features to instruments, and making traffic controllers that automatically sense traffic load and adjust the duration of the signals, etc. The MCS-40 microcomputer system offers the designer a way to add significant features to systems at trivial costs.

Reduced Complexity

Because the MCS-40 system eliminates many ICs and consequently the failures associated with these devices, it can significantly increase system reliability. Most of the failures in a digital system occur because an interconnect has failed. The use of a typical 16 pin IC will introduce approximately 36 interconnections in a system. There are 16 interconnections from the chip to the lead frame, 16 from the lead frame to the PC card, and approximately 2 interconnections from the PC card to the back plane, and 2 interconnections from back plane point to back plane point per IC. If one ROM eliminates fifty ICs, then it eliminates approximately 1800 interconnections.

Conclusion

The MCS-40 microcomputer has reduced the cost of putting basic computation into a device by a factor of 10 or more. As such it can bring to many new systems the benefits of using computers. Because of its small size and small cost, the MCS-40 system can be designed into many devices such as cash registers, scales, stop lights, instruments, etc., where the use of a computer was once unthinkable.

The benefits of putting an MCS-40 microcomputer into a system go far beyond the advantages of merely being able to include computation or decision making into the device being designed. As previously indicated, the use of an MCS-40 system can affect such basic things as manufacturing cost, market share, development costs and time, and system reliability and serviceability.

The following list summarizes the MCS-40 System Components' major features:

4040 — Central Processor Unit

- Instructions (60 total) including Logical Operations and Read Program Memory
- Large number of family devices
- 10.8 microsecond instruction cycle standard
- Interrupt capability
- Single step operation
- 8K byte memory addressing capability and up to 5120 bits of RAM
- 24 index registers
- Subroutine nesting to 7 levels
- 2-phase dynamic operation
- Instruction set includes conditional branching, jump to subroutine and indirect fetching
- Logical instructions
- Binary and decimal arithmetic modes
- Unlimited number of input and output lines

4004 — Central Processor Unit

- 4 bit parallel CPU with 46 instructions
- Can be used with standard memory components
- 2-phase dynamic operation
- Instruction set includes conditional branching, jump to subroutine and indirect fetching
- Binary and decimal arithmetic modes
- Unlimited number of input and output lines
- 10.8 microsecond instruction cycle standard
- 4K byte memory addressing capability
- 16 index registers

4201 – Clock Generator

- Provides 2-phase clock – TTL and MOS
- Crystal controlled oscillator
- Directly drives MCS-40™ set
- Generates power on reset for MCS-40
- Provides single step circuit for 4040

MEMORIES

4308 – Mask Programmable ROM

- 1K x 8 program storage
- Four independent 4 bit I/O ports
- Buffered inputs and outputs
- Equivalent of four 4001 ROMs (256 x 8)
- Directly TTL Compatible

4001 – Mask Programmable ROM

- 256 x 8 program storage
- 4 bit I/O port
- Directly TTL Compatible

4002 – RAM

- 320 bits organized 4 x 80
- Output 4 bit port
- Directly TTL compatible

INPUT/OUTPUT DEVICES

4265 – Programmable General Purpose I/O

- 14 programmable operating modes
- 16 lines of I/O capability
- Bit set/reset capability
- Interfaces to 8080 peripherals
- Synchronous and asynchronous interfaces
- Directly TTL compatible

4269 – Programmable Keyboard/Display Device

- Independent interfaces to keyboard input and display output
- Keyboard input for sensor matrix, scanned keyboard, or encoded keyboard
- Display output for individually scanned display devices (up to 16 characters) or Self-Scan* displays (up to 20 characters).
- 8 character keyboard FIFO
- Roll-over and debounce control

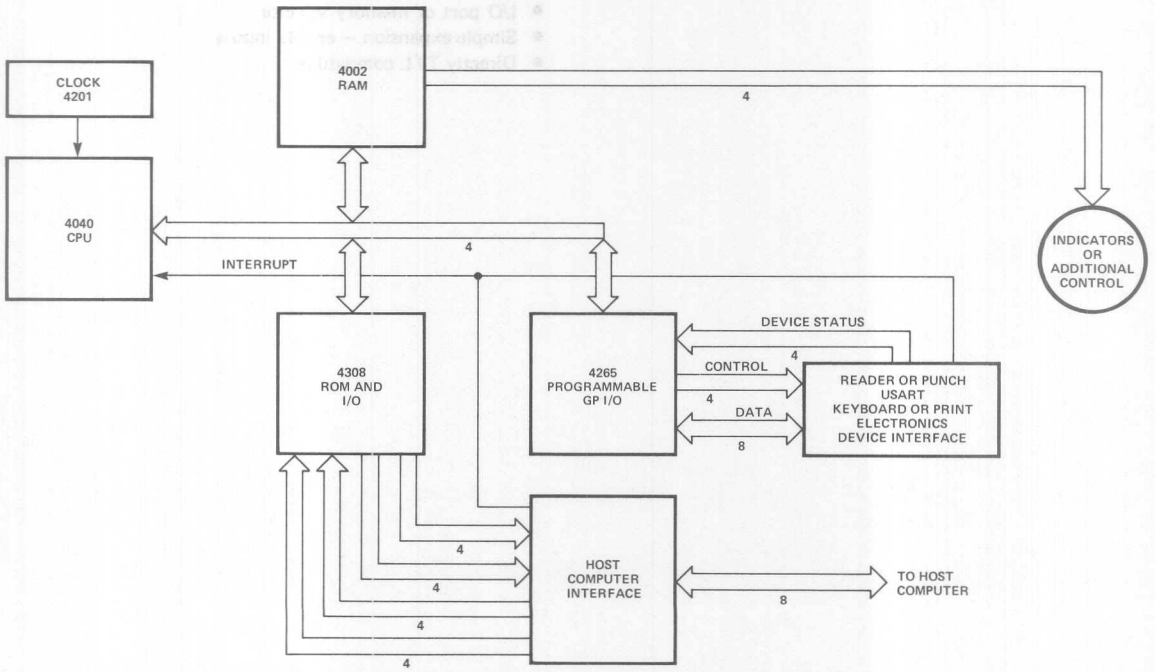
*Self-Scan is a registered trademark of Burroughs Corporation.

4003 – Shift Register

- 10 bit serial in/parallel out
- Serial-output allows for expansion
- Asynchronous clock

4289 – Standard Memory Interface

- Direct interface to all standard memories: TTL, NMOS, PMOS, CMOS
- Allows READ and WRITE program memory
- Unlimited Input/Output expansion
- Single package equivalent of 4008/4009



Typical MCS-40 Application – Peripheral I/O Controller.

SUPPLEMENTAL DEVICES

The following devices are supplemental and are compatible with the 4289.

4702A — Erasable and Electrically Reprogrammable ROM

- 2048 bits, organized 256 x 8
- Fast programming — 2 minutes for 2048 bits
- Inputs and outputs TTL compatible
- Three-state outputs
- Alterable Program Memory in system development

4316A — Mask Programmable ROM

- 16,384 bits, organized 2048 x 8
- Fully decoded

2101 RAM

- 1024 bits — organized 256 x 4
- Static operation
- Fully decoded
- Directly TTL compatible
- Three-state outputs
- Used for writeable Program Memory

5101 CMOS RAM

- Utra low standby current: 15nA/bit
- Static operation
- 1024 bits — organized 256 x 4

8-BIT I/O DEVICES

The following are 8-bit devices which interface to an MCS-40™ CPU via the 4265 Programmable General Purpose I/O.

8251 — Programmable Serial Communications Device

- Synchronous and asynchronous operation
- Baud rate — DC to 56K Baud (Synchronous Mode)
  — DC to 9.6K Baud (Asynchronous Mode)
- Full duplex, double buffered transmitter and receiver
- Error detection — parity, overview, and framing

8253 — Programmable Interval Timer

(Available 2nd Quarter 1976)

- Three independent 16-bit counters
- Programmable counter modes
- Counts binary or BCD

PERIPHERAL DEVICES

3214 — Priority Interrupt Control Unit

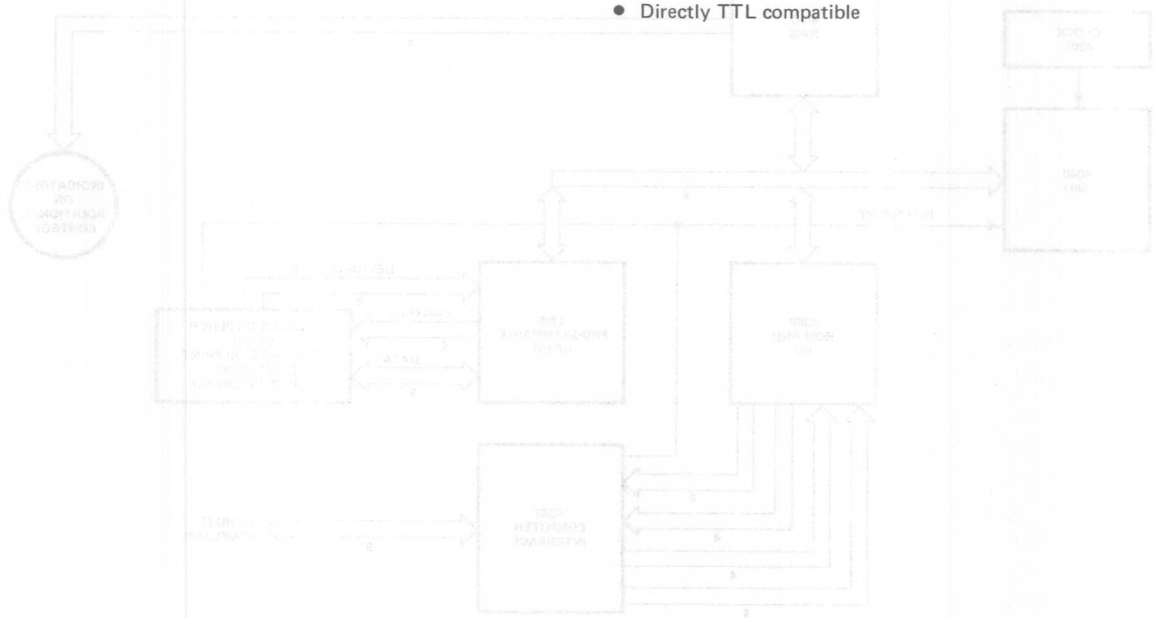
- Eight priority levels
- Current status register
- Priority comparator

3216/3226 — 4-Bit Bidirectional Bus

- Low input load current
- High output drive
- Three-state outputs
- Inverted (3226) and non-inverted (3216) operations

3205 — High Speed 1 out of 8 Binary Decoder

- I/O port or memory selector
- Simple expansion — enable inputs
- Directly TTL compatible



THE FUNCTIONS OF A COMPUTER

This section introduces certain basic computer concepts. It provides background information and definitions which will be useful in later sections of this manual. Those already familiar with computers may skip this material, at their option.

A Typical Computer System

A typical digital computer consists of:

a. A central processor unit (CPU)
b. A memory
c. Input/Output (I/O) ports

The program memory serves primarily as a place to store *instructions,* the coded pieces of data that direct the activities of the CPU. A group of logically related instructions stored in memory is referred to as a *program.* The CPU "reads" each instruction from memory in a logically determinate sequence, and uses it to initiate processing actions. If the program structure is coherent and logical, processing produces intelligible and useful results.

The data memory is used to store the data to be manipulated. The CPU can access any data stored in memory; but often the memory is not large enough to store the entire data bank required for a particular application. The problem can be resolved by providing the computer with one or more *input ports.* The CPU can address these ports and input the data contained there. The addition of input ports enables the computer to receive information from external equipment (such as a paper tape reader) at high rates of speed and in large volumes.

Almost any computer requires one or more *output ports* that permit the CPU to communicate the result of its processing to the outside word. The output may go to a display, for use by a human operator, to a peripheral device that produces "hard-copy", such as a line-printer, to a peripheral storage device, such as a magnetic tape unit, or the output may constitute process control signals that direct the operations of another system, such as an automated assembly line. Like input ports, output ports are addressable. The input and output ports together permit the processor to interact with the outside world.

The CPU unifies the system. It controls the functions performed by the other components. The CPU must be able to fetch instructions from memory, decode their binary contents and execute them. It must also be able to reference memory and I/O ports as necessary in the execution of instructions. In addition, the CPU should be able to recognize and respond to certain external control signals, such as INTERRUPT and STOP requests. The functional units within a CPU that enable it to perform these functions are described below.

The Architecture of a CPU

A typical central processor unit (CPU) consists of the following interconnected functional units:

1. Registers
2. Arithmetic/Logic Unit (ALU)
3. Control Circuitry

Registers are temporary storage units within the CPU. Some registers, such as the program counter and instruction register, have dedicated uses. Other registers, such as the accumulator, are for more general purpose use.

ACCUMULATOR:

The accumulator usually stores one of the operands to be manipulated by the ALU. A typical instruction might direct the ALU to add the contents of some other register to the contents of the accumulator and store the result in the accumulator itself. In general, the accumulator is both a source (operand) and destination (result) register.

Often a CPU will include a number of additional general purpose registers that can be used to store operands or intermediate "scratch-pad" data.

PROGRAM COUNTER (JUMPS, SUBROUTINES AND THE STACK):

The instructions that make up a program are stored in the system's memory. The central processor examines the contents of the memory, in order to determine what action is appropriate. This means that the processor must know which location contains the next instruction.

Each of the locations in memory is numbered, to distinguish it from all other locations in memory. The number which identifies a memory location is called its *address.*

The processor maintains a counter which contains the address of the next program instruction. This register is called

the *program counter.* The processor updates the program counter by adding "1" to the counter each time it fetches an instruction, so that the program counter is always current.

The programmer therefore stores his instructions in numerically adjacent addresses, so that the lower addresses contain the first instructions to be executed and the higher addresses contain later instructions. The only time the programmer may violate this sequential rule is when the last instruction in one block of memory is a *jump* instruction to another block of memory.

A jump instruction contains the address of the instruction which is to follow it. The next instruction may be stored in any memory location, as long as the programmed jump specifies the correct address. During the execution of a jump instruction, the processor replaces the contents of its program counter with the address embodied in the jump. Thus, the logical continuity of the program is maintained.

A special kind of program jump occurs when the stored program accesses or "branches" to a subroutine. In this kind of jump, the processor is logically required to "remember" the contents of the program counter at the time that the jump occurs. This enables the processor to resume execution of the main program when it is finished with the last instruction of the subroutine.

A *subroutine* is a program within a program. Usually it is a general-purpose set of instructions that must be executed repeatedly in the course of a main program. Routines which calculate the square, the sine, or the logarithm of a program variable are good examples of the functions often written as subroutines. Other examples might be programs designed for inputting or outputting data to a particular peripheral device.

The processor has a special way of handling subroutines, in order to insure an orderly return to the main program. When the processor receives a jump to subroutine instruction, it increments the program counter and stores the counter's contents in a register memory area known as the *stack.* The stack thus saves the address of the instruction to be executed after the subroutine is completed. Then the processor stores the address specified in the subroutine jump in its program counter. The next instruction fetched will therefore be the first step of the subroutine.

The last instruction in any subroutine is a branch back. Such an instruction need specify no address. When the processor fetches a branch back instruction, it simply replaces the current contents of the program counter with the address on the top of the stack. This causes the processor to resume execution of the program at the point immediately following the original branch.

Subroutines are often *nested;* that is, one subroutine will sometimes call a second subroutine. The second may call a third, and so on. This is perfectly acceptable, as long as the processor has enough capacity to store the necessary return addresses, and the logical provision for doing so. In other words, the maximum depth of nesting is determined by the depth of the stack itself. If the stack has space for storing three return addresses, then three levels of subroutines may be accommodated.

Processors have different ways of maintaining stacks. Most have facilities for the storage of return addresses built into the processor itself. The integral stack is usually more efficient, since fewer steps are involved in the execution of a call or a return.

INSTRUCTION REGISTER AND DECODER:

Every computer has a *word length* that is characteristic of that machine. A computer's word length is usually determined by the size of its internal storage elements and interconnecting paths (referred to as *buses*); for example, a computer whose registers and buses can store and transfer 4 bits of information has a characteristic word length of 4 bits and is referred to as a 4 bit parallel processor.

The characteristic eight bit field is sometimes referred to as a *byte*, a four bit field can be referred to as a *nibble.*

Each operation that the processor can perform is identified by a unique binary number known as an *instruction code.* An eight bit word used as an instruction code can distinguish among 256 alternative actions, more than adequate for most processors.

The processor fetches an instruction in two distinct operations. In the first, it transmits the address in its program counter to the memory. In the second, the memory returns the addressed byte to the processor. The CPU stores this instruction byte in a register known as the *instruction register,* and uses it to direct activities during the remainder of the instruction execution.

The mechanism by which the processor translates an instruction code into specific processing actions requires more elaboration than we can here afford. The concept, however, will be intuitively clear to any experienced logic designer. The eight bits stored in the instruction register can be decoded and used to selectively activate one of a number of output lines, in this case up to 256 lines. Each line represents a set of activities associated with execution of a particular instruction code. The enabled line can be combined coincidentally with selected timing pulses, to develop electrical signals that can then be used to initiate specific actions. This translation of code into action is performed by the *instruction decoder* and by the associated control circuitry.

An eight bit field is more than sufficient, in most cases, to specify a particular processing action. There are times, however, when execution of the instruction code requires more information than eight bits can convey.

One example of this is when the instruction references a memory location. The basic instruction code identifies the operation to be performed, but cannot specify the object address as well. In a case like this, a two-word instruction must be used. Successive instruction bytes are stored in sequentially adjacent memory locations, and the processor performs two fetches in succession to obtain the full instruction. The first byte retrieved from memory is placed in the processor's instruction register, and the second byte is placed in temporary storage, as appropriate. When the entire instruction is fetched, the processor can proceed to the execution phase.

ADDRESS REGISTER(S):

A CPU may use a register or register-pair to temporarily store the address of a memory location that is to be accessed for data. If the address register is *programmable,* (i.e., if there

are instructions that allow the programmer to alter the contents of the register) the program can "build" an address in the address register prior to executing a *memory reference* instruction (i.e., an instruction that reads data from memory, writes data to memory or operates on data stored in memory).

ARITHMETIC/LOGIC UNIT (ALU):

All processors contain an arithmetic/logic unit, which is often referred to simply as the ALU. By way of analogy, the ALU may be thought of as a super adding machine with its keys commanded automatically by the control signals developed in the instruction decoder and the control circuitry. This is essentially how the first stored-program digital computer was conceived.

The ALU naturally bears little resemblance to a desk-top adder. The major difference is that the ALU calculates by creating an electrical analogy, rather than by mechanical analogy. Another important difference is that the ALU uses binary techniques – rather than decimal methods – for representing and manipulating numbers. In principle, however, it is convenient to think of the ALU as an electronically controlled calculator.

The ALU must contain an *adder* which is capable of combining the contents of two registers in accordance with the logic of binary arithmetic. This provision permits the processor to perform arithmetic manipulations on the data it obtains from memory and from its other inputs.

Using only the basic adder a capable programmer can write routines which will subtract, multiply and divide, giving the machine complete arithmetic capabilities. In practice, however, most ALUs provide other built-in functions, including hardware subtraction, boolean logic operations, and shift capabilities.

The ALU contains *flag bits* which register certain conditions that arise in the course of arithmetic manipulations. Flags typically include *carry* and *zero*. It is possible to program jumps which are conditionally dependent on the status of one or more flags. Thus, for example, the program may be designed to jump to a special routine, if the carry bit is set following an addition instruction.

CONTROL CIRCUITRY:

The control circuitry is the primary functional unit within a CPU. Using clock inputs, the control circuitry maintains the proper sequence of events required for any processing task. After an instruction is fetched and decoded, the control circuitry issues the appropriate signals (to units both internal and external to the CPU) for initiating the proper processing action. Often the control circuitry will be capable of responding to external signals, such as an interrupt request. An *interrupt* request will cause the control circuitry to temporarily interrupt main program execution, jump to a special routine to service the interrupting device, then automatically return to the main program.

COMPUTER OPERATIONS

There are certain operations that are basic to almost any computer. A sound understanding of these basic operations is a necessary prerequisite to examining the specific operations of a particular computer.

TIMING:

The activities of the central processor are cyclical. The processor fetches an instruction, performs the operations required, fetches the next instruction, and so on. An orderly sequence of events like this requires timing, and the CPU therefore contains a free running oscillator clock which furnishes the reference for all processor actions. The combined fetch and execution of a single instruction is referred to as an *instruction cycle.* The portion of a cycle identified with a clearly defined activity is called a phase. And the interval between pulses of the timing oscillator is referred to as a *clock period.* As a general rule, one or more clock periods are necessary to the completion of a phase, and there are several phases in a cycle.

INSTRUCTION FETCH:

The first phase(s) of any instruction cycle will be dedicated to fetching the next instruction. The CPU issues a read operation code and the contents of the program counter are sent to program memory, which responds by returning the next instruction word. The first word of the instruction is placed in the instruction register. If the instruction consists of more than one word, additional cycles are required to fetch each word of the instruction. When the entire instruction is present in the CPU, the program counter is incremented (in preparation for the next instruction fetch) and the instruction is decoded. The operation specified in the instruction will be executed in the remaining phases of the instruction cycle. The instruction may call for a memory read or write, an input or output and/or an internal CPU operation, such as a register-to-register transfer or an add-registers operation.

MEMORY READ:

The instruction fetched may then call for data to be read from data memory into the CPU. The CPU issues a read operation code and sends the proper memory address; memory responds by returning the requested word. The data received is placed in the accumulator (not the instruction register).

MEMORY WRITE:

A program memory write operation is similar to a read except for the direction of data flow. The CPU issues a write operation code, sends the proper memory address, then sends the data word to be written into the addressed memory location.

INPUT/OUTPUT:

Input and Output operations are similar to memory read and write operations with the exception that a peripheral I/O port is addressed instead of a memory location. The CPU issues the appropriate input or output command, sends the

proper device address and either receives the data being input or sends the data to be output.

Data can be input/output in either parallel or serial form. All data within a digital computer is represented in binary coded form. A binary data word consists of a group of bits; each bit is either a one or a zero. *Parallel* I/O consists of transferring all bits in the word at the same time, one bit per line. *Serial* I/O consists of transferring one bit at a time on a single line. Naturally serial I/O is much slower, but it requires fewer signal paths.

INTERRUPTS:

Interrupt provisions are included on many central processors, as a means of improving the processor's efficiency. Consider the case of a computer that is processing a large volume of data, portions of which are to be output to a printer. The CPU can output a byte of data within a single instruction cycle but it may take the printer the equivalent of many instruction cycles to actually print the character specified by the data byte. The CPU will have to remain idle waiting until the printer can accept the next data byte. If an interrupt capability is implemented on the computer, the CPU can output a data byte then return to data processing. When the printer is ready to accept the next data byte, it can request an interrupt. When the CPU acknowledges the interrupt, it suspends main program execution and automatically branches to a routine that will output the next data byte. After the byte is output, the CPU continues with main program execution. Note that this is, in principle, quite similar to a subroutine call, except that the jump is initiated externally rather than by the program.

More complex interrupt structures are possible, in which several interrupting devices share the same processor. Interruptive processing is an important feature that enables maximum utilization of a processor's capacity.

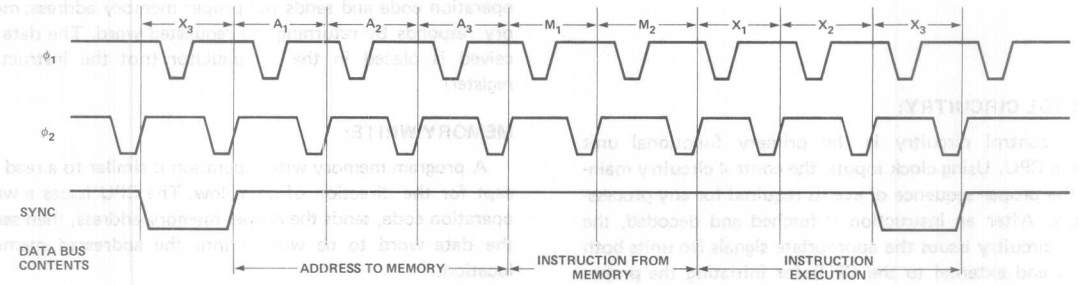
MCS-40™ SYSTEM OPERATION

The MCS-40 microcomputer, like all computer systems, contains a processor and memory. The processor (4040,4004) is contained in one dual in-line I.C. package. The memory can be either ROM program memory or RAM data memory. In addition to information exchanges between the processor and memory, the processor also provides input and output (I/O) to external devices. The I/O facilities of the MCS-40 system are found in a separate I/O device or shared with memory elements, allowing for a reduction in package count. All MCS-40 inter-element communication transpires over a four bit data bus (D_0-D_3), D_3 being the most significant bus bit.

The data bus transfers information such as instruction addresses, operand addresses, operands, and I/O data from the processor to the memory and I/O elements. The processor receives instructions, operands, and I/O data back from the other elements. All traffic on the bus is contained within one instruction cycle for one cycle instructions. The instruction cycle is subdivided into equal time segments. Each segment is equivalent to one system clock period. Information on the data bus is altered from segment to segment. The first three time segments present a twelve bit (three groups of four bits) instruction address to the memory, least significant nibble first. The fourth and fifth segments provide the 8 bit instruction sequentially placed on the data bus by program memory. The sixth segment is utilized for instruction decode. The remaining two segments are used for program execution. Operands and I/O data can be found on the data bus during this time, depending on the instruction being executed.

Instructions requiring two cycles proceed in a similar manner as described above. Complete execution requires 16 clock periods which are partitioned into two eight segment cycles. During the first cycle the instruction will be fetched. Information fetched during this first cycle may also include a portion of the operand or an indirect address register. The second cycle will always fetch the operand and perform the execution.

Differentiation between one and two cycle instructions are performed by all elements on the data bus by decoding the unique instructions.



MCS-40™ Basic Timing Diagram

CHAPTER 1. PROCESSORS

MCS-40™ CENTRAL PROCESSOR UNITS

The MCS-40 family consists of two powerful central processor architectures, that of the 4040, and its predecessor, the 4004. The 4004 instruction set is a subset of the 4040 instruction set and programs generated for the 4004 will operate on the 4040. Hence, the 4040 is electrically and functionally upward compatible with the 4004. This mutual compatibility is true for all family components.

The 4004 will be described first, followed by a description of the 4040 as an extension of the 4004. The last section of this chapter is a detailed description of the instruction format and operation. The reader may find it useful to reference this section when reading the component descriptions.

The user will find the 4004/4040 Assembly Language Programming Manual useful as additional reference material.

NOTE: The MCS-40 MOS components are implemented with negative logic where a logic "1" or true is represented by the most negative voltage (referred to as V_{DD}) and a logic "0" or false is represented by the most positive voltage (referred to as V_{SS}). In the cases where positive logic conventions are used for interfacing with TTL devices, the positive logic conventions will be explicitly stated.

4004 CENTRAL PROCESSOR

Introduction

The 4004 is a central processor unit designed to work in conjunction with the other members of the MCS-40 microcomputer set to form a completely self-contained system. The CPU communicates with the other members of the set through a four line data bus and with the user peripheral devices through the RAM, ROM, GP I/O or special purpose I/O ports. The CPU chip contains 5 command control lines, four of which are used to control the RAM chips (each line can control up to 4 RAM chips for a total system capacity of 16 RAM's) and one which is used to control a ROM bank of up to 4K words of program memory.

Instructions

The instruction repertoire of the 4004 consists of:

a. 16 machine instructions (5 of which are double length)

b. 14 accumulator group instructions (Decimal/Binary operation)

c. 16 input/output instructions

The instruction set and its format will be described in detail in a subsequent section.

Hardware Description

The 4004 is packaged in a 16 pin DIP. The pin configuration is shown in the following figure. A brief functional description of each pin is given below:

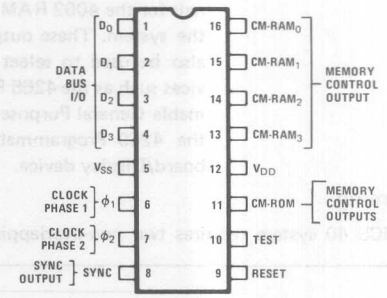


Figure 1-1. 4004 Pin Configuration.

Pin Description

Pin No.	Designation	Description of Function
1-4	D_0-D_3	Bidirectional data bus. All address and data communication between the processor and the RAM and ROM chips is handled by way of these 4 lines.
5	V_{SS}	Most positive supply voltage.
6-7	ϕ_1-ϕ_2	Non-overlapping clock signals which determine processor timing
8	SYNC	SYNC output. Synchronization signal generated by the processor and sent to ROM and RAM chips. Indicates beginning of instruction cycle.

Pin No.	Designation	Description of Function
9	RESET	RESET input. A "1" level (V_{DD}) applied to this pin clears all flag and status flip-flops and forces the program counter to 0. To completely clear all of the address and index registers, RESET must be applied for 64 clock cycles (8 instruction cycles).
10	TEST	TEST input. The logical state of this input can be examined with the JCN instruction.
11	CM-ROM	Command-ROM output. This pin enables a ROM bank, which can contain up to 4K words of program using 4001, 4308, or 4289. It can also be used to enable I/O devices which are attached to the CM-ROM.
12	V_{DD}	Main supply voltage to the processor. Value must be V_{SS} -15.0V ±5%.
13-16	$CM\text{-}RAM_0$-$CM\text{-}RAM_3$	Command-RAM outputs. These outputs act as bank select signals for the 4002 RAM chips in the system. These outputs can also be used to select I/O devices such as the 4265 Programmable General Purpose I/O and the 4269 Programmable Keyboard/Display device.

Basic Timing

The MCS 40 system requires two non-overlapping clock phases, ϕ_1, ϕ_2 which are supplied by the 4201 clock generator. The 4004 will generate a SYNC signal every 8 clock periods and will send it to the other system components. The SYNC signal marks the beginning of each instruction cycle. The other MCS-40™ components will use the SYNC, ϕ_1, and ϕ_2 signals to generate internal timing.

A typical machine cycle starts with the CPU sending a synchronization signal (SYNC) to the ROM's and RAM's. Next, 12 bits of ROM address are sent to the data bus using three clock periods. The address is then incremented by one and stored in the program counter. The selected ROM sends back 8 bits of instruction or data during the following 2 clock periods. This information is stored in two registers: OPR and OPA. The next three clock periods are used to execute the instruction. (See Basic Instruction Cycles in Figure 1-2.)

Basic Description of Major Circuit Blocks

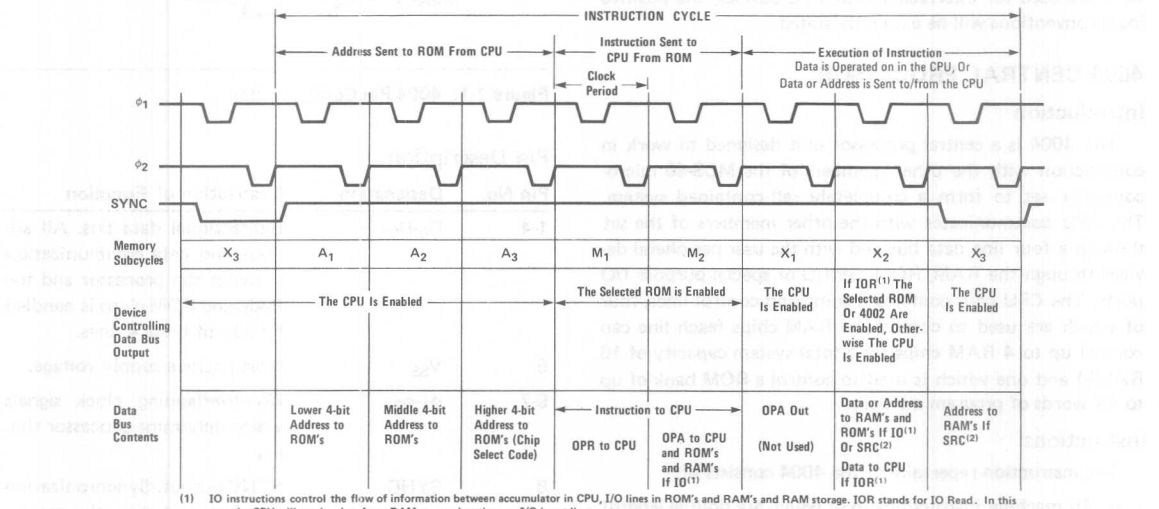
The 4004 block diagrams shown in Figures 1-3 and 1-4 contain the following functional blocks:

1. Address register (program counter and stack organized as 4 words of 12 bits each) and address incrementer.
2. Index register (64 bits organized as 16 words of 4 bits each).
3. 4 bit adder.
4. Instruction register (8 bits wide), decoder and control.
5. Peripheral circuitry.

The functional blocks communicate internally through a 4-line bus. The function and composition of each block is as follows:

1. Address Register (Program Counter & Stack) & Address Incrementer

The address register is a dynamic RAM cell array of 4 x 12 bits. It contains one level used to store the instruction



(1) IO instructions control the flow of information between accumulator in CPU, I/O lines in ROM's and RAM's and RAM storage. IOR stands for IO Read. In this case the CPU will receive data from RAM storage locations or I/O input lines.

(2) The SRC instruction designates the chip number and address for a following IO instruction.

Figure 1-2. 4004 Basic Instruction Cycle.

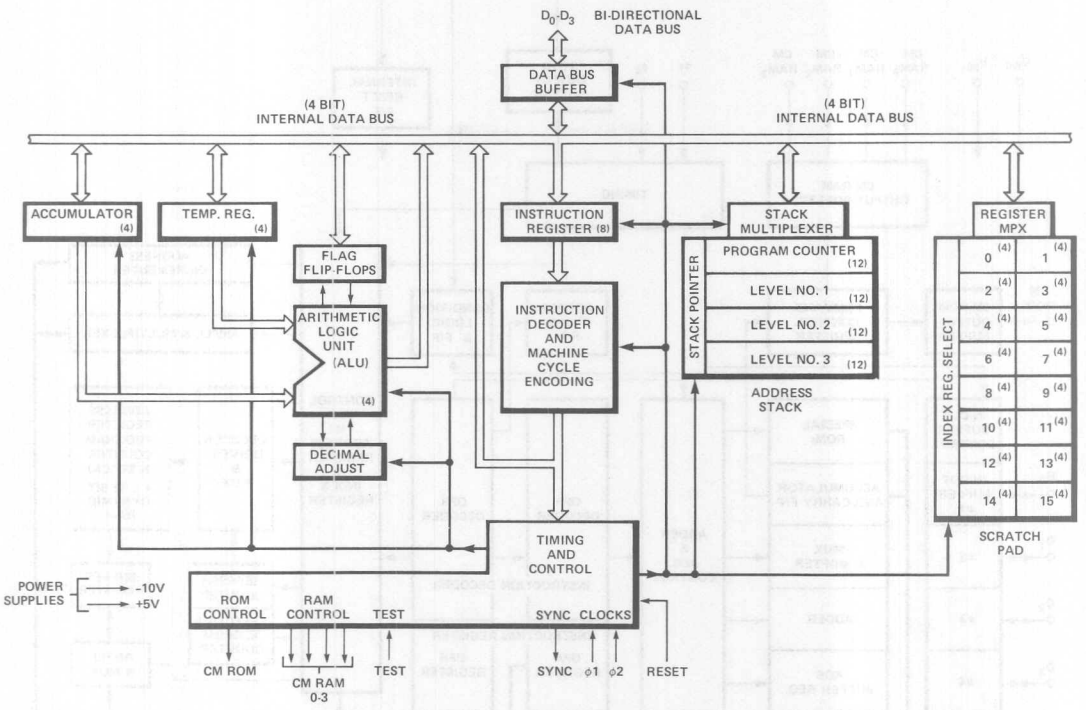


Figure 1-3. 4004 CPU Block Diagram.

address (program counter) and 3 levels used as a stack for subroutine calls. The stack address is provided by the effective address counter and by the refresh counter, and it is multiplexed to the decoder.

The address when read is stored in an address buffer and is demultiplexed to the internal bus during A_1, A_2, and A_3 in three 4 bit nibbles. The address is incremented by a 4 bit carry look-ahead circuit (address incrementer) after each 4 bit nibble is sent out on the data bus. The incremented address is transferred back to the address buffer and finally written back into the address register.

2. Index Register

The index register is a dynamic RAM cell array of 16 x 4 bits and has two modes of operation. In one mode of operation the index register provides 16 directly addressable storage locations for intermediate computation and control. In the second mode, the index register provides 8 pairs of addressable storage locations for addressing RAM and ROM as well as for storing data fetched from ROM.

The index register address is provided by the internal bus and by the refresh counter and is multiplexed to the index register decoder.

The content of the index register is transferred to the internal bus through a multiplexer. Writing into the register is accomplished by transferring the content of the internal bus into a temporary register and then to the index register.

3. 4 Bit Adder

The 4 bit adder is of the ripple-through carry type. One term of the addition comes from the "ADB" register which communicates with the internal bus on one side and can transfer data or $\overline{\text{data}}$ to the adder. The other term of the addition comes from the accumulator and carry flip-flop. Both data and $\overline{\text{data}}$ can be transferred. The output of the adder is transferred to the accumulator and carry FF. The accumulator is provided with a shifter to implement rotate right and rotate left instructions. The accumulator also communicates with the command control register, special ROM's, the condition flip-flop and the internal bus. The command control register holds a 3 bit code used for CM-RAM line switching. The special ROM's perform a code conversion for DAA (decimal adjust accumulator) and KBP (Keyboard Process) instructions. The special ROM's also communicate with the internal bus. The condition logic senses adder = 0 and ACC = 0 conditions, the state of the carry FF, and the state of an external signal (TEST) to implement JCN (jump on condition) and ISZ (increment index register skip if zero) instructions.

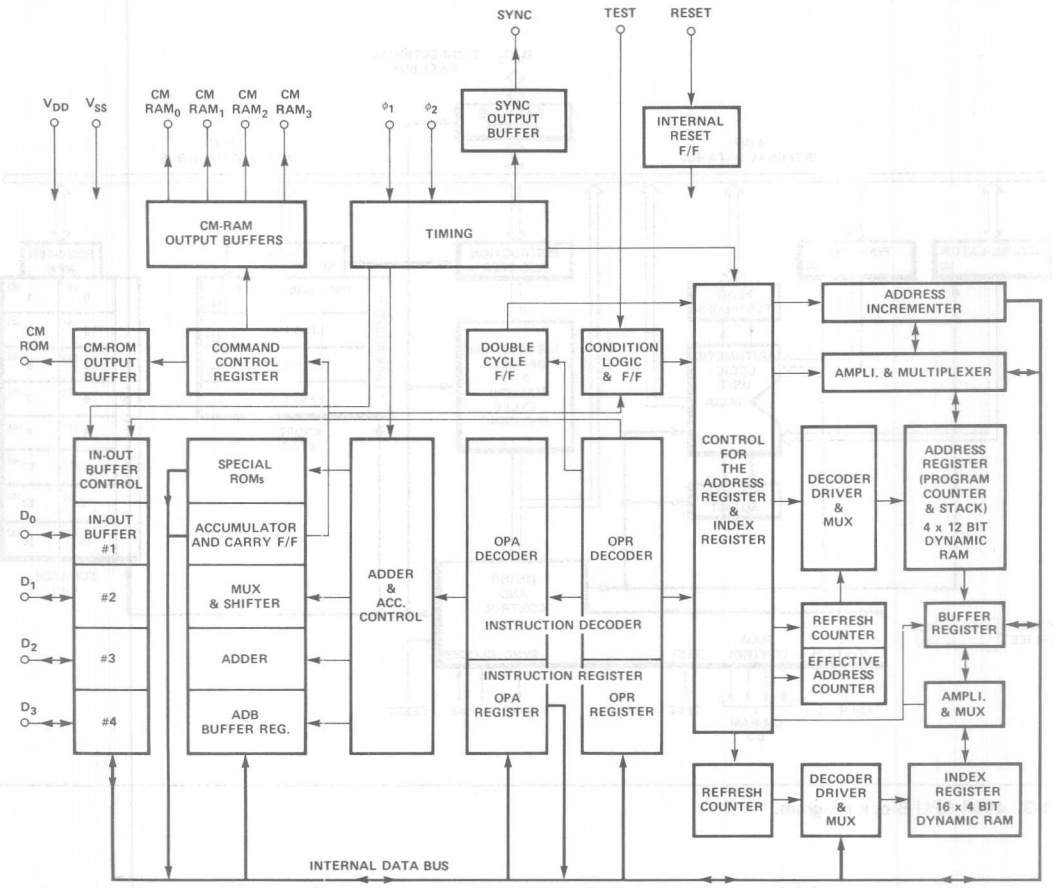


Figure 1-4. 4004 Detailed Block Diagram.

4. Instruction Register Decoder and Control

The instruction register (consisting of the OPR Register and OPA Register each 4 bits wide) is loaded with the contents of the internal bus (at M_1 and M_2 time in the instruction cycle) through a multiplexer and holds the instruction fetched from ROM. The instructions are decoded in the instruction decoder and appropriately gated with timing signals to provide the control signals for the various functional blocks. A Double Cycle FF is set from any one of 5 double-length instructions. Double-length instructions are instructions whose length is 16 bits wide (instead of 8 bits) and that require two system cycles (16 clock cycles) for their execution. Double length instructions are stored in two successive locations in ROM. A condition FF controls JCN and ISZ instructions and is set by the condition logic. The state of an external pin "test" can control one of the conditions in the JCN instruction.

5. Peripheral Circuitry

This includes:

a. The data bus input-output buffers communicating between data paths and internal bus.
b. Timing and SYNC generator.
c. 1 ROM command control (CM-ROM) and the 4 RAM command control ($CM\text{-}RAM_i$) output buffers.
d. Reset flip-flop.

During reset (Reset pin low), all RAM's and static FF's are cleared, and the data bus is set to "0" (V_{SS}). After reset, program control will start from address "0" and $CM\text{-}RAM_0$ is selected. To completely clear all registers and RAM locations in the CPU the reset signal must be applied for at least 8 full instruction cycles (64 clock cycles) to allow the index register refresh counter to scan all locations in memory. (256 clock cycles for the 4002 RAM.)

Applying reset to the CPU initiates the reset operations regardless of the particular instruction cycle state in which the CPU was in at the time of the reset signal being lowered.

4040 CENTRAL PROCESSOR

Introduction

The 4040 is a single chip 4 bit parallel MOS central processor. It is intended as an enhanced version of the 4004 and as such retains all of the functional capability of that device. It does, however, provide several significant improvements in hardware and software. These are listed briefly here and will be described in detail in the body of this specification.

Extended Instruction Set

The 4040 software contains all of the 4004 instruction set and includes an additional 14 instructions, providing:

- Halt
- Logical operations
- Interrupt disable, enable functions
- ROM Bank switching
- Index register bank switching

This instruction set is described in detail in Definition of Instruction Set section.

Additional Features

The 4040 contains the necessary hardware to accept and process single level interrupts. The interrupt jumps the program to location 003 while saving some key processor conditions.

The address stack has been increased from 4 x 12 bits to 8 x 12 bits, allowing up to seven levels of subroutine nesting.

The index register array has been increased from sixteen 4 bit registers to twenty-four 4 bit registers.

The 4040 is provided with a STOP control which allows the user to halt the processor at an instruction cycle. This feature allows the implementation of a 'single step' operation for program debugging (see STOP/HALT Mode Operation section below).

The 4040 can address up to 8K x 8 words of ROM with no external logic required. This is implemented by having two 4K x 8 memory banks that can be toggled between.

More Flexible Interface and System Configurations

The 4040 is provided with separate power supply pins for the timing circuitry and for the output buffers. These features allow a low-power standby mode by shutting off the main power supply and operating only the timing. Since the output buffers have a separate supply they can be directly interfaced to other circuit types such as N-channel MOS or CMOS. For single-supply systems all three power supply pins can be tied together.

Hardware Description

The 4040 is packaged in a 24 pin DIP. The pin configuration is shown in the following figure. A brief functional description of each pin is given in the following Pin Description.

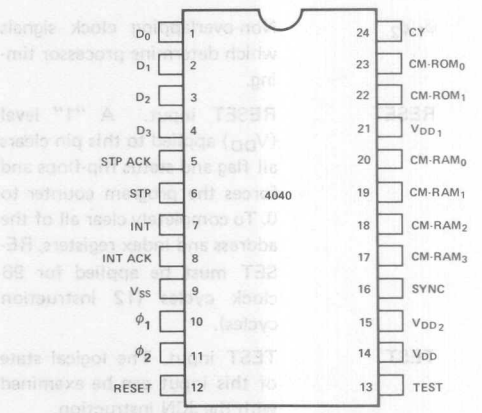


Figure 1-5. 4040 Pin Configuration.

Pin Description

Pin No.	Designation	Description of Function
1-4	D_0-D_3	Bidirectional data bus. All address and data communication between the processor and the RAM and ROM chips is handled by way of these 4 lines. D_0 is the least significant bit and corresponds to accumulator A_0.
5	STPA†	STOP ACKNOWLEDGE output. A negative level (V_{DD}) on this signal acknowledges that the processor has entered the stop mode. Output is "open drain" requiring pull-down resistor to V_{DD}.
6	STP*	STOP input signal. A negative level at this input causes the processor to enter the STOP mode.
7	INT*	INTERRUPT input signal. A negative level at this input causes the processor to enter the INTERRUPT mode.
8	INTA†	INTERRUPT ACKNOWLEDGE output. This signal acknowledges receipt of an INTERRUPT command and prevents additional INTERRUPTs from entering the processor. INTERRUPT ACKNOWLEDGE remains active negative until cleared by the BRANCH BACK and SRC (BBS) instruction. The output is "open drain", requiring a pull-down resistor to V_{DD}.
9	V_{SS}	Most positive supply voltage.
10-11	ϕ_1-ϕ_2	Non-overlapping clock signals which determine processor timing.
12	RESET	RESET input. A "1" level (V_{DD}) applied to this pin clears all flag and status flip-flops and forces the program counter to 0. To completely clear all of the address and index registers, RESET must be applied for 96 clock cycles (12 instruction cycles).
13	TEST*	TEST input. The logical state of this input can be examined with the JCN instruction.
14	V_{DD}	Main supply voltage to the processor. Value must be V_{SS} -15.0V ±5%.
15	V_{DD2}	Supply voltage for output buffers. May be varied depending on interface conditions. This line controls D_0-D_3, CM-RAM_{0-3}, CM-$ROM_{0,1}$ and SYNC.
16	SYNC	SYNC output. Synchronization signal generated by the processor and sent to ROM and RAM chips. Indicates beginning of instruction cycle.
17-20	CM-RAM_0-CM-RAM_3	CM-RAM outputs. These outputs act as bank select signals for the 4002 RAM chips in the system. The outputs can also be used to select I/O devices such as the 4265 Programmable I/O and the 4269 Programmable Keyboard Display device.
21	V_{DD1}	Supply voltage for timing circuit. Value must be V_{SS} -15.0V ±5%. Allows low power standby operation. Only SYNC will be generated but not outputted when this pin is the only active V_{DD}.
22-23	CM-ROM_1 CM-ROM_0	CM-ROM outputs. These outputs act as bank select signals for the ROM chips in the system. These outputs can also be used to enable I/O devices which are attached to CM-ROM_O or CM-ROM_1.
24	CY†	CARRY output buffer. The state of the CY flip-flop is presented at this output and is updated at X_1. The output is "open drain" requiring a pull-down resistor to V_{DD}.

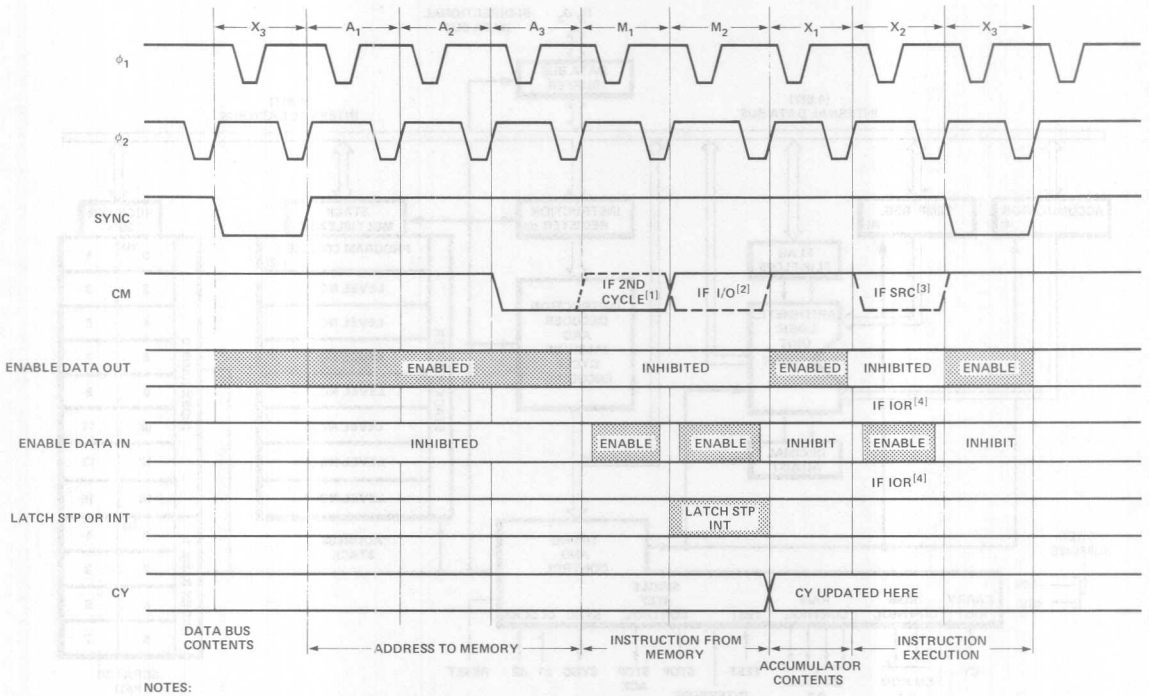
* This signal may be driven by a TTL output with a 1K pull-up resistor to V_{SS}.

† This signal may drive a low power TTL input using a 12K pulldown resistor to V_{DD}.

Basic Circuit Timing

The basic system timing for the 4040 is identical to that used for the 4004, as shown in the following figure. Two non-overlapping clock signals, ϕ_1 and ϕ_2, are used to define the basic timing. The start of an instruction cycle is indicated by the SYNC signal, which is generated by the processor and sent to the various ROM and RAM or peripheral chips in the system. An instruction cycle consists of the following operations:

1. The 12 bit content of the program counter is sent out to the ROM chips in three 4 bit groups during A_1, A_2, A_3.



NOTES:

1. CM-ROM, RAM SIGNALS WILL BE LOW (V_{DD}) AT M_1 FOR ANY SINGLE CYCLE INSTRUCTION OR FOR THE FIRST CYCLE OF A DOUBLE CYCLE INSTRUCTION.
2. CM-ROM, RAM SIGNALS WILL BE LOW (V_{DD}) AT M_2 FOR ANY OF THE SIXTEEN I/O GROUP INSTRUCTIONS.
3. CM-ROM, RAM SIGNALS WILL BE LOW (V_{DD}) AT X_2 DURING EXECUTION OF AN SRC INSTRUCTION.
4. IOR MEANS ONE OF THE I/O READ INSTRUCTIONS: SBM, ROM, RDR, ADM, RDϕ, RD1, RD2, RD3.

Figure 1-6. 4040 Basic Timing Diagram.

2. The 8 bit instruction or data from the addressed ROM location is received by the processor at M_1 and M_2 at which time the instruction is decoded.
3. Instruction execution occurs during X_1, X_2, and X_3. Data or address information may be sent to output ports or RAM chips; data may be received from input ports or RAM chips; or data may be operated on within the processor.

The data bus contents at the various times of the instruction cycle are defined just as for the 4004 with the exception of the data at X_1 and the carry output during X_3 of a No-Op instruction. The 4040 outputs the contents of the accumulator at X_1 for program debugging purposes, whereas the 4004 simply copies the data which it received at M_2. The data bus contents at X_2 and X_3 depend on the instruction being executed; a listing for each individual instruction is contained in the Data Bus Activity section.

A timing feature not present in the 4004 occurs with the generation of the CM-ROM, CM-RAM signals at M_1. This will occur for all single cycle instructions and for the first cycle of all double cycle instructions. This feature allows external logic to distinguish between instruction information and address or data at M_1 and M_2 time.

Basic Description of Major Circuit Blocks

The following figure is a block diagram of the 4040 indicating the major circuit blocks and their interconnections. The following major functional blocks are contained in the 4040:

1. Address register stack and address incrementer.
2. Index register array.
3. 4 bit adder/accumulator.
4. Instruction register/decoder and control logic.
5. Hardware interrupt and stop control.
6. Peripheral circuits for controlling timing and external communication.

A brief functional description of each of these major elements is given below.

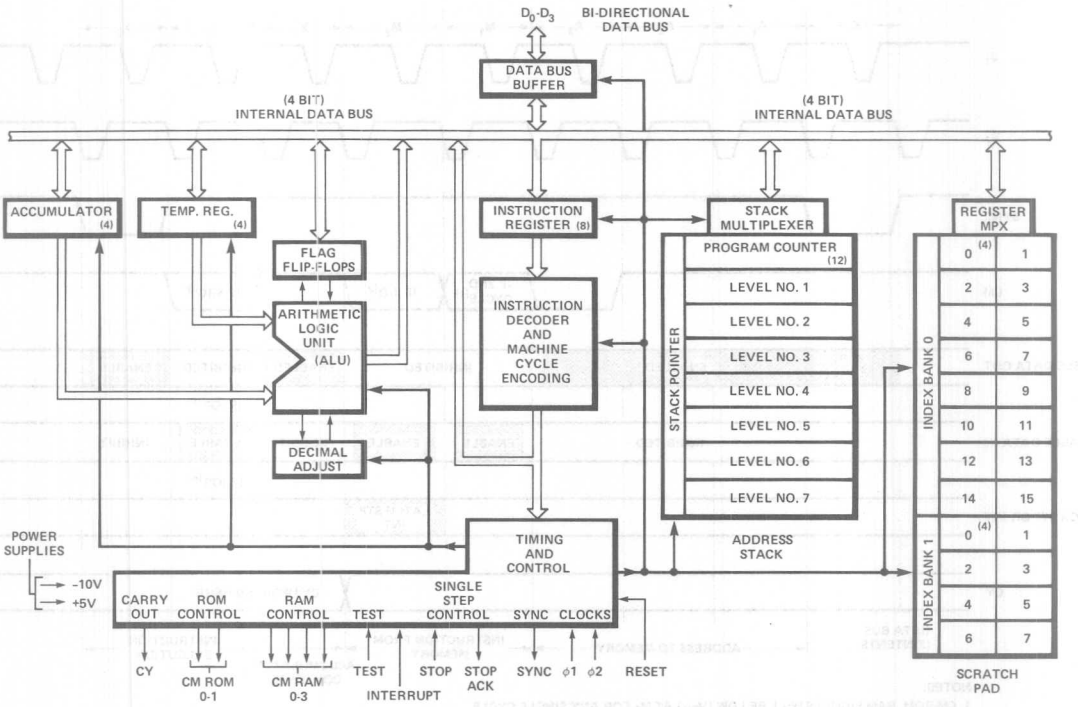


Figure 1-7. 4040 CPU Block Diagram.

Address Register and Address Incrementer

The address register is a dynamic RAM array of 8 x 12 bits operating as a push-down stack. One level of the stack is used to store the instruction address, leaving seven levels available for subroutine calls and interrupt processing. The stack address is provided by the effective address counter to the decoder.

The contents of the selected address register are stored in the address buffer and multiplexed to the internal bus during A_1, A_2, and A_3 in 4 bit nibbles. The contents of the address buffer are incremented by a 4 bit carry-look-ahead circuit following the outputting of each 4 bit nibble. The incremented value is transferred back to the address buffer and written back into the selected address register.

Since the array is dynamic, provision is made for refreshing the stored data. A 3 bit refresh counter is multiplexed to the stack decoder for this purpose.

Index Register Array

The index register is a dynamic RAM array of 12 x 8 bits organized as three banks of 4 x 8 bits. Two of the banks have identical address locations and so must be individually selected with the SB0, SB1 instructions. The third bank is always available for use. Refer to the description in the Expanded Index Register Array organization in the next section.

Two modes of operation are possible for the index register array. In one mode the array provides 24 directly addressable 4 bit storage locations for intermediate computation or control purposes. In the second mode the array provides 12 pairs of register locations for addressing RAM, ROM and I/O ports or for storing data fetched from ROM.

Index register addressing is provided by the internal bus for normal read/write operations and by a refresh counter for refresh operation. The addresses are multiplexed to the array decoder.

The content of the selected register is stored in a temporary register and multiplexed to the internal bus. During write operations the internal bus contents are transferred to the temporary register and then to the selected index register.

SRC Save Register

The SRC save register is an 8 bit dynamic latch which stores the contents of the designated index register pair during the execution of the SRC instruction. This 8 bit value is sent to the ROM and RAM chips as an address for any succeeding I/O instruction (see detailed description in Definition of Instruction Set section). The SRC save register is used to hold this value in the case that an interrupt should occur, thus allowing the value to be automatically restored when a re-

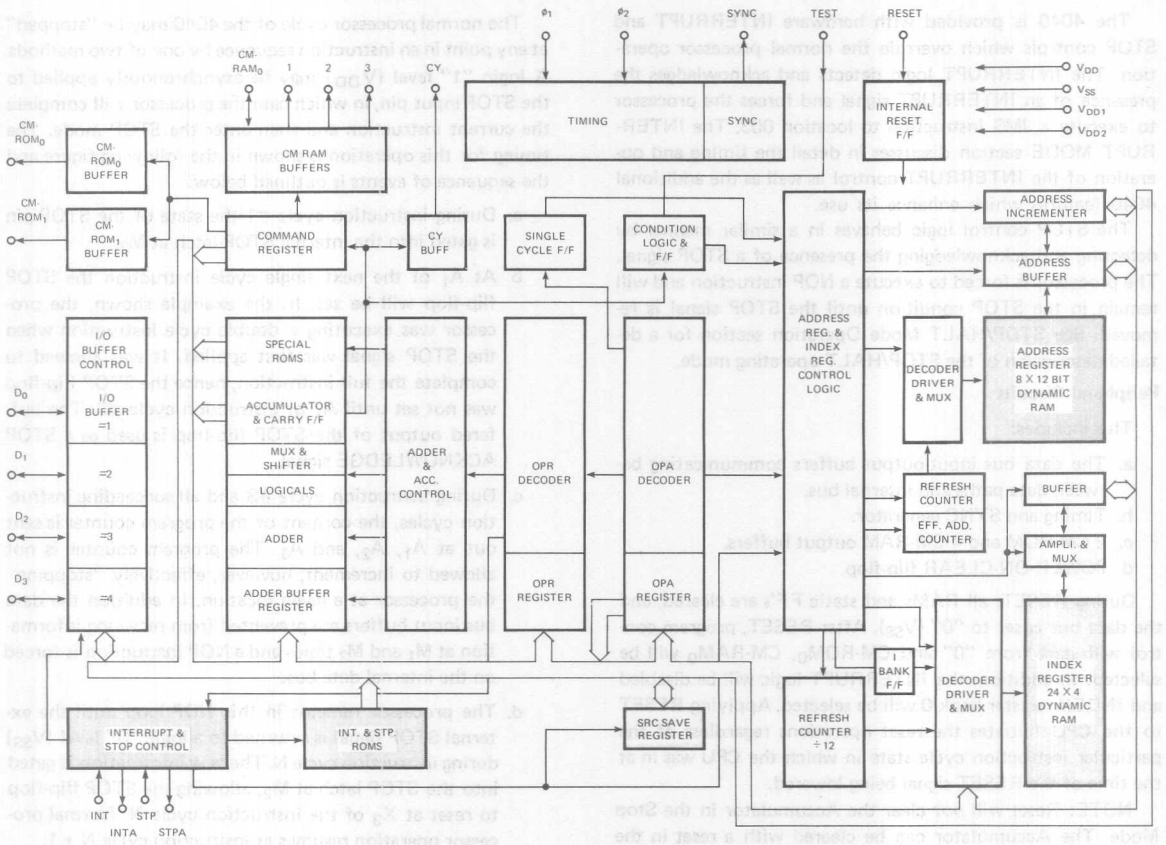


Figure 1-8. Detailed 4040 Block Diagram.

turn from interrupt is made. During an interrupt routine, SRC instructions executed do not affect the value stored in the SRC save register.

4 Bit Adder/Accumulator

The 4 bit adder is of the ripple-through carry type. One term of the addition comes from the "Adder Buffer" register which communicates with the internal bus on one side and can transfer to the adder data or $\overline{\text{data}}$. The other term of the addition comes from the accumulator and carry flip-flop. Both data and $\overline{\text{data}}$ can be transferred. The output of the adder is transferred to the accumulator and carry FF. The accumulator is provided with a shifter to implement shift right and shift left instructions. The accumulator communicates also with the command register, with special ROMs, with the condition logic, and with the internal bus. The command register holds a 3 bit code used for CM-RAM line switching and one bit for CM-ROM switching. The special ROMs perform a code conversion for DAA (decimal adjust accumulator) and KBP (process keyboard) instructions. The special ROMs communicate with the internal bus. The condition logic senses adder = 0 and ACC = 0 conditions, the state of the carry FF, and the state of an external signal (TEST) to implement JCN (jump on condition) and ISZ (increment index register skip if zero) instructions.

Instruction Register/Decoder and Control Logic

The instruction register is loaded with the content of the internal bus at M_1 and M_2 during first instruction cycle through a multiplexer and holds the instruction fetched. The instructions are decoded in the instruction decoder and appropriately gated with timing signals to provide the control signals for the various functional blocks. A single cycle FF is reset from one of 5 double-length instructions. Double-length instructions are instructions that need two instruction cycles (16 clock periods) for their execution. A condition FF controls JCN and ISZ instruction and is set by the condition logic.

INTERRUPT and STOP Control Logic

The 4040 is provided with hardware INTERRUPT and STOP controls which override the normal processor operation. The INTERRUPT logic detects and acknowledges the presence of an INTERRUPT signal and forces the processor to execute a JMS instruction to location 003. The INTERRUPT MODE section discusses in detail the timing and operation of the INTERRUPT control as well as the additional 4040 features which enhance its use.

The STOP control logic behaves in a similar manner by detecting and acknowledging the presence of a STOP signal. The processor is forced to execute a NOP instruction and will remain in the STOP condition until the STOP signal is removed. See STOP/HALT Mode Operation section for a detailed description of the STOP/HALT operating mode.

Peripheral Circuits

This includes:

a. The data bus input-output buffers communicating between data paths and internal bus.
b. Timing and SYNC generator.
c. 2 CM-ROM and 4 CM-RAM output buffers.
d. POWER-ON-CLEAR flip-flop.

During RESET, all RAMs and static FF's are cleared, and the data bus is set to "0" (V_{SS}). After RESET, program control will start from "0" and CM-ROM$_0$, CM-RAM$_0$ will be selected. In addition, the INTERRUPT logic will be disabled and INDEX register bank 0 will be selected. Applying RESET to the CPU initiates the reset operations regardless of the particular instruction cycle state in which the CPU was in at the time of the RESET signal being lowered.

NOTE: Reset will *not* clear the Accumulator in the Stop Mode. The Accumulator can be cleared with a reset in the Run Mode.

4040 Unique Operating Features

The following features will be described in detail:

- STOP/HALT mode logic
- INTERRUPT mode logic
- Extended ROM addressing capability

STOP/HALT Mode Operation

The normal processor cycle of the 4040 may be "stopped" at any point in an instruction sequence by one of two methods. A logic "1" level (V_{DD}) may be asynchronously applied to the STOP input pin, in which case the processor will complete the current instruction and then enter the STOP mode. The timing for this operation is shown in the following figure and the sequence of events is outlined below:

a. During instruction cycle #1 the state of the STOP pin is gated into the internal STOP latch at M_2.

b At A_1 of the next single cycle instruction the STOP flip-flop will be set. In the example shown, the processor was executing a double cycle instruction when the STOP signal was first applied. It was allowed to complete the full instruction, hence the STOP flip-flop was not set until A_1 of instruction cycle #3. The buffered output of the STOP flip-flop is used as a STOP ACKNOWLEDGE signal.

c. During instruction cycle #3 and all succeeding instruction cycles, the content of the program counter is sent out at A_1, A_2, and A_3. The program counter is not allowed to increment, however, effectively "stopping" the processor at a given location. In addition the data bus input buffers are prevented from receiving information at M_1 and M_2 times and a NOP instruction is forced on the internal data bus.

d. The processor remains in this NOP loop until the external STOP signal is returned to a logic "0" level (V_{SS}) during instruction cycle N. The new information is gated into the STOP latch at M_2, allowing the STOP flip-flop to reset at X_3 of the instruction cycle N. Normal processor operation resumes at instruction cycle N + 1.

HALT Mode

Entry to the STOP mode may also be gained through the use of the HALT (HLT) instruction as shown in the following figure. In this case the processor executes the HLT instruction and causes the HALT and STOP flip-flops to be set at X_3 of instruction cycle #1. The processor is forced to execute

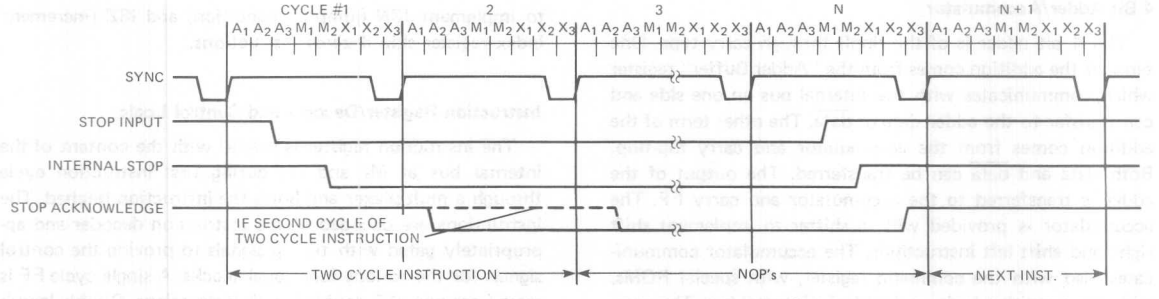


Figure 1-9. Stop Timing.

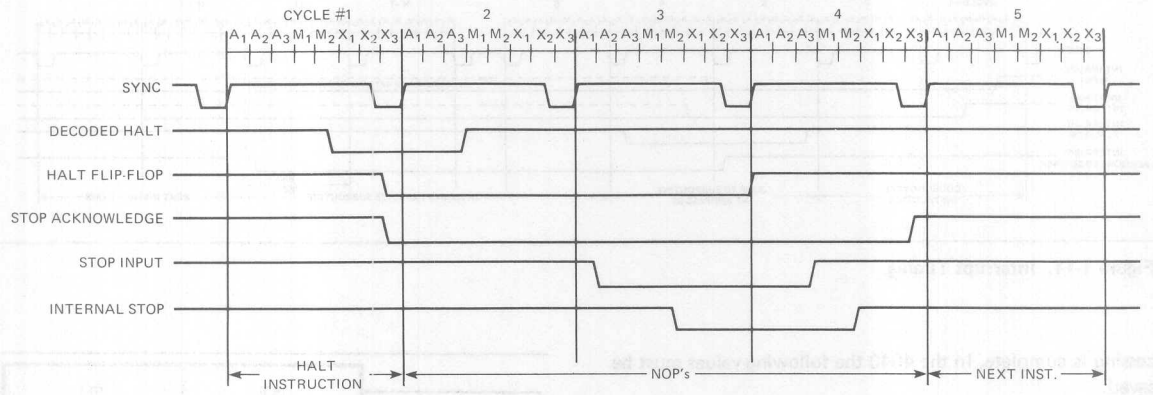


Figure 1-10. Halt Timing (Exit Using Stop Input).

NOP's at instruction cycle #2 and all successive cycle times until removed from the HALT mode.

Exit from the HALT mode can be gained in two ways, one of which is shown in the following figure. At instruction cycle #N, a logic "1" level (V_{DD}) is applied to the STOP input and is in turn latched by the STOP latch at M_2. A logic "1" (V_{DD}) in the STOP latch causes the HALT flip-flop to be reset at A_1 of cycle #N + 1. The processor is now in the normal STOP mode and can be released as described in (d) above.

The second means of exiting from the HALT condition is by way of the INTERRUPT input and will be described in the INTERRUPT Mode section.

DATA BUS Contents

The data bus contents during STOP/HALT mode are shown. For program debugging purposes the following information is available:

A_1, A_2, A_3	12 bit address from internal program counter.
M_1, M_2	8 bit instruction from addressed ROM location. Internally the processor executes NOP.
X_1	4 bit contents of ACCUMULATOR.
X_2, X_3	8 bit contents of internal SRC save register which stores the value of the last SRC address. CM-ROM and CM-RAM signals are not present at X_2 in this case. (See INTERRUPT Mode section for complete description of operation of SRC register.)

Single Step Operation

The STOP control provides a convenient means of program debugging by allowing a "single step" operation whereby the 4040 can be stepped instruction-by-instruction under the control of a push-button switch.

INTERRUPT Mode

The 4040 is provided with an asynchronous INTERRUPT input and an INTERRUPT ACKNOWLEDGE output. The following figure presents the basic timing for the INTERRUPT mode. The sequence of events is as follows:

a. During instruction cycle #1 an INTERRUPT occurs and is gated into the INTERRUPT LATCH during M_2.

b. At A_1 of the next single cycle instruction the INTERRUPT flip-flop is set. As in the case of the STOP example, if the processor is executing a double cycle instruction it is allowed to complete it.

c. During instruction cycle #3 the program counter is prevented from incrementing and the data input buffers are inhibited at M_1 and M_2. A JUMP TO SUBROUTINE (JMS) instruction is forced on the internal data bus. The subroutine address is forced to be location 3. At X_3 the INTERRUPT ACKNOWLEDGE flip-flop is set and its buffered output is available on the INTERRUPT ACKNOWLEDGE pin. The instruction at location 3 begins the interrupt processing routine.

d. The INTERRUPT ACKNOWLEDGE flip-flop remains set until the interrupt has been processed and the BRANCH BACK and SRC (BBS) instruction has been executed (instruction cycle #N). No new INTERRUPT can be entered while INTERRUPT ACKNOWLEDGE is active. Note that the INTERRUPT signal may be removed after INTERRUPT ACKNOWLEDGE occurs.

Saving and Restoring Processor Status

To have an effective interrupt handling capability the processor must be capable of saving current program and status register values and restoring same when the interrupt pro-

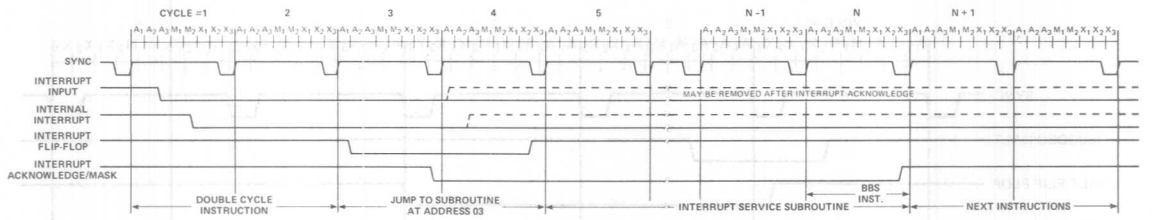


Figure 1-11. Interrupt Timing.

cessing is complete. In the 4040 the following values must be saved:

a. Content of ACCUMULATOR and CARRY flip-flop.
b. Content of COMMAND REGISTER.
c. Content of as many INDEX REGISTERS as required.
d. The value of the last SRC address sent out prior to interrupt.
e. Content of the PROGRAM COUNTER.
f. The current ROM bank (CM-ROM$_0$ or CM-RAM$_1$).

To facilitate the items listed, a number of new hardware features have been included in the 4040 and are described in the following paragraphs.

Expanded Index Register Array

Saving status values requires having temporary storage locations available in the index register array. For this reason the 4040 is provided with 8 additional index registers (4 register pairs) providing a total of 24 4 bit registers (12 8 bit register pairs). The array is organized into three 8 register banks. Bank 0 and Bank 1 are individually selectable by using the SELECT INDEX BANK (SB0, SB1) instructions. The upper bank, which contains registers 8 - 15, is always available for storage. The Interrupt Processing section in Chapter 2 provides examples of interrupt routines and demonstrates the use of the index register banks as well as all other features listed here.

Note that both Bank 0 and Bank 1 contain the same individual address locations. This feature allows those instructions which reference specific register locations to be executed from either index register bank. Thus the JUMP INDIRECT (JIN) instruction and the logical instructions OR4, OR5, AN6, AN7 reference two different sets of registers, e.g., a JIN instruction can reference register pair #0 in Bank 0 or in Bank 1.

The BANK SELECT flip-flop is automatically saved and restored during interrupts. Thus a user may wish to operate in Bank 0 until interrupted then switch to Bank 1. When the BBS instruction is executed to return from interrupt, the previous Bank will automatically be selected for the next instruction.

After application of a RESET signal, Bank 0 will be selected.

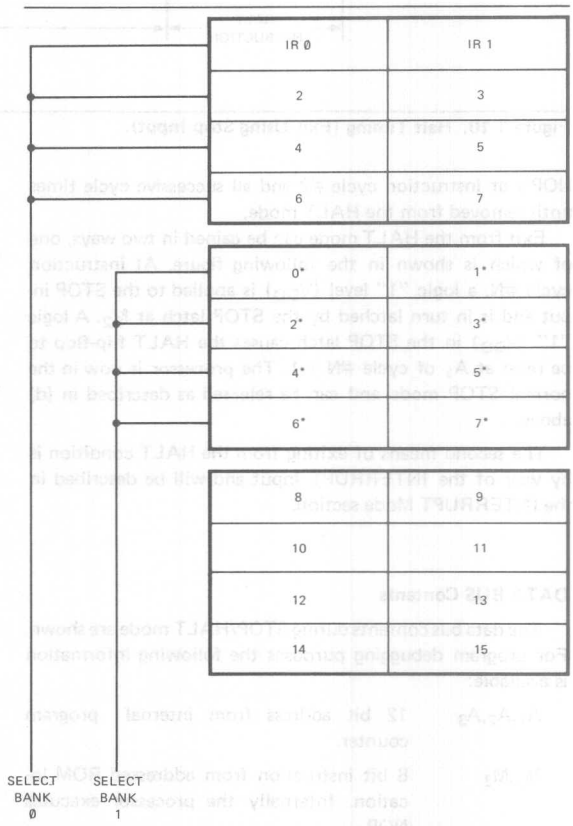


Figure 1-12. Index Register Organization.

SRC Save Register

When the 4040 executes an SRC instruction, the 4 bit values sent out at X_2 and X_3 are stored internally in the 8 bit SRC register. The SRC save register is locked out during the interrupt routine by the INTERRUPT ACKNOWLEDGE flip-flop. Thus, any SRC instruction executed during an interrupt routine will not affect the value in the SRC save register. The last instruction in the interrupt routine must be a BRANCH BACK and SRC (BBS), an

instruction which restores the program counter (see below) and sends out the contents of the SRC save register at X_2 and X_3, and generates the appropriate CM-ROM and CM-RAM at X_2. This restores the ROM and RAM select logic to their pre-interrupt conditions.

Extended Address Register Stack

The address stack of the 4040 is an 8 x 12 bit array (compared to 4 x 12 bits in the 4004). One level of the stack is required for the program counter; hence seven levels of subroutines may be nested using the 4040.

When an interrupt occurs the program counter is not incremented, but is stored down one level in the stack by the execution of the forced JMS instruction. This value is restored by the execution of the BBS instruction at the end of the interrupt routine.

General Information Applying to Interrupt

The 4040 is capable of servicing one interrupt at a time. The INTERRUPT ACKNOWLEDGE signal is used internally to prevent a second interrupt from being entered until the first is completely serviced.

Two instructions, INTERRUPT ENABLE (EIN) and INTERRUPT DISABLE (DIN) are provided for protecting sequences of instructions from being interrupted. These are described in detail in the Definition of Instruction Set section.

The RESET signal disables interrupt. If the processor is started from a RESET condition, an EIN instruction must be performed before an interrupt will be recognized.

If an INTERRUPT and STOP signal occur such that they are both latched at M_2 of the same instruction cycle, the STOP logic will have priority, and must be cleared before the interrupt can be recognized.

As mentioned in the DATA BUS Contents section, the INTERRUPT control may be used to exit from a HALT condition. The timing for this is shown. The processor enters the HALT mode at instruction cycle #1 and remains in that mode until the INTERRUPT signal is recognized at instruction cycle #4. When the INTERRUPT flip-flop is set, it causes the HALT flip-flop to be reset. The processor is then in the INTERRUPT mode. In this way a processor could be used in a completely asynchronous control application in which the INTERRUPT signal would begin the processing routine. When the routine was complete the processor would execute the HLT instruction and wait for a new INTERRUPT.

Extended ROM Addressing Capability

The 4040 is equipped with two CM-ROM output buffers, each of which can be used to select a bank of sixteen 256 x 8 words. A total of 8K x 8 bit words can be directly addressed. Bank switching is accomplished through the use of two instructions, DESIGNATE BANK 0 (DB0) and DESIGNATE BANK 1 (DB1). Both of these instructions take effect on the third cycle following their execution. The Use of DB0 & DB1 section in Chapter 2 provides example uses of the DB0, 1 instructions.

Since the INTERRUPT control logic will force a JMS to location 3, the first few instructions of the interrupt routine will have to be duplicated in both ROM banks (see Interrupt Processing section).

The fact that the bank switching operation requires three instruction cycles to be completed means that an INTERRUPT, Stop and Halt cannot occur during those three cycles. For this reason the INTERRUPT Stop and Halt logic is internally disabled during the execution of DB0 or DB1.

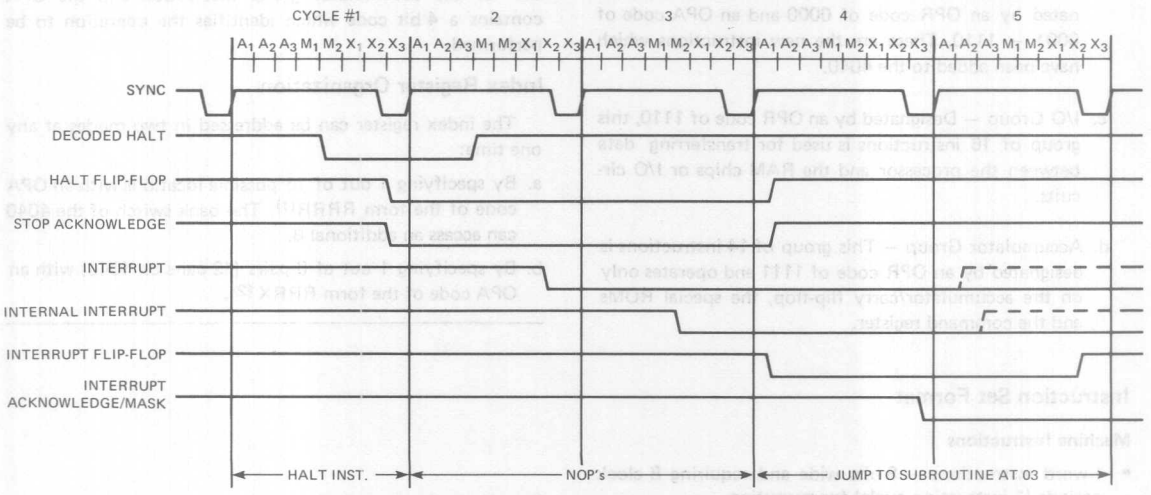


Figure 1-13. Halt Timing (Exit Using Interrupt).

MCS-40™ PROGRAM MEMORY ORGANIZATION

MCS-40 program memory is normally located in a Read Only Memory (ROM) such as a 4001 ROM or a 4308 ROM. By the use of interface components such as the 4289 Standard Memory Interface, MCS-40 program memory can be in a PROM memory (Programmable Read Only Memory) such as a 4702A PROM component or a RAM memory such as a 4101 RAM component. Program memory is always organized as an 8-bit wide memory array.

MCS-40 program memory is also conceptually divided into 256-byte pages. A page of memory is by definition a group of 256 bytes where the high-order 4 bits of the 12-bit address are the same. For 4001 ROMs, which are 256 x 8 memory components, a page is physically contained in one ROM chip. Certain MCS-40 instructions address data or other instructions only within the page in which the instruction itself is located. In other words, the 12-bit address of data or another instruction is formed by combining the 4 high-order bits from the program counter (the current page number) with the remaining 8 bits of address provided by the instruction.

INSTRUCTION SET

The 4040 is functionally compatible with the 4004 and therefore recognizes all 46 instructions valid for the 4004. In addition, the 4040 recognizes 14 new instructions giving a total of 60 instructions in the set. The instruction format is, of course, identical to that used in the 4004.

Four groups of instructions can be defined as follows:

a. Machine Instructions – This group of 16 instructions are designated by an OPR code of 0000 – 1101. Within this group is contained a second group which is designated supplemental group.

b. 4040 Group – This group of 14 instructions is designated by an OPR code of 0000 and an OPA code of 0001 – 1110. These are the new instructions which have been added to the 4040.

c. I/O Group – Designated by an OPR code of 1110, this group of 16 instructions is used for transferring data between the processor and the RAM chips or I/O circuits.

d. Accumulator Group – This group of 14 instructions is designated by an OPR code of 1111 and operates only on the accumulator/carry flip-flop, the special ROMs and the command register.

Instruction Set Format

Machine Instructions

- 1-word instructions – 8 bits wide and requiring 8 clock periods (1 instruction cycle) for execution
- 2-word instructions – 16 bits wide and requiring 16 clock periods (2 instruction cycles) for execution

A 1-word instruction occupies one location in ROM (each location can hold one 8 bit word) and a 2-word instruction occupies two successive locations in ROM. Each instruction word is divided into two 4 bit nibbles. The upper 4 bits is called the OPR and contains the operation code. The lower 4 bits is called the OPA and contains the modifier. For a single word machine instruction the operation code (OPR) contains the code of the operation that is to be performed (add, subtract, load, etc.). The modifier (OPA) contains one of 4 things:

1. A register address
2. A register pair address
3. 4 bits of data
4. An instruction modifier

For a 2-word machine instruction the first word is similar to a 1-word instruction, however, the modifier (OPA) contains one of 4 things:

1. A register address
2. A register pair address
3. The upper portion of another ROM address
4. A condition for jumping

The 2nd word contains either the middle portion (in OPR) and lower portion (in OPA) of another ROM address or 8 bits of data (the upper 4 bits in OPR and the lower 4 bits in OPA).

The upper 4 bits of instruction (OPR) will always be fetched before the lower 4 bits of instruction (OPA) during M_1 and M_2 times respectively.

Input/Output & RAM Instructions and Accumulator Group Instructions

In these instructions (which are all single word) the OPR contains a 4 bit code which identifies either the I/O instruction or the accumulator group instruction and the OPA contains a 4 bit code which identifies the operation to be performed.

Index Register Organization

The index register can be addressed in two modes at any one time:

a. By specifying 1 out of 16 possible locations with an OPA code of the form RRRR(1). The bank switch of the 4040 can access an additional 8.

b. By specifying 1 out of 8 pairs (12 pairs on 4040) with an OPA code of the form RRRX (2).

Note 1: In this case, the instruction is executed on the 4-bit contents addressed by RRRR.
Note 2: In this case the instruction is executed on the 8-bit content addressed by RRRX, where X is 0 or 1 depending on the instruction.

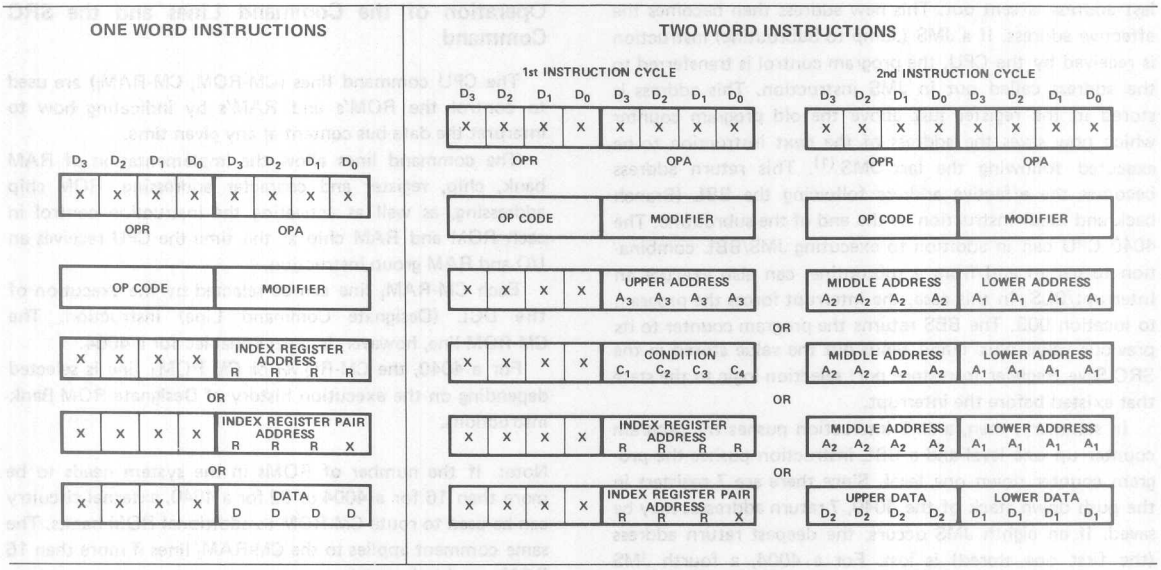


Figure 1-14. Machine Instruction Format.

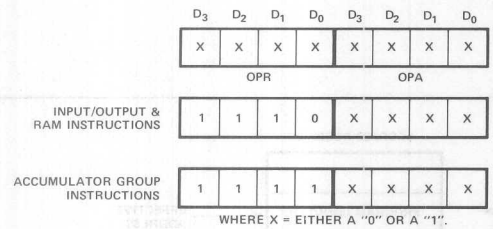


Figure 1-15. I/O and Accumulator Group Instruction Formats.

When the index register is used as a pair register, the even number register (RRR0) is used as the location of the middle address or the upper data fetched from the ROM, the odd number register (RRR1) is used as the location of the lower address or the lower data fetched from the ROM.

Operation of the Address Register (Program Counter and Stack)

The address register contains four 12 bit registers for the 4004 and eight 12 bit registers for the 4040; one register is used as the program counter and stores the instruction address; the other registers make up the push down stack.

Initially, any one of the address registers can be used as the program counter to store the instruction address. In a typical sequence, the program counter is incremented by 1 after the

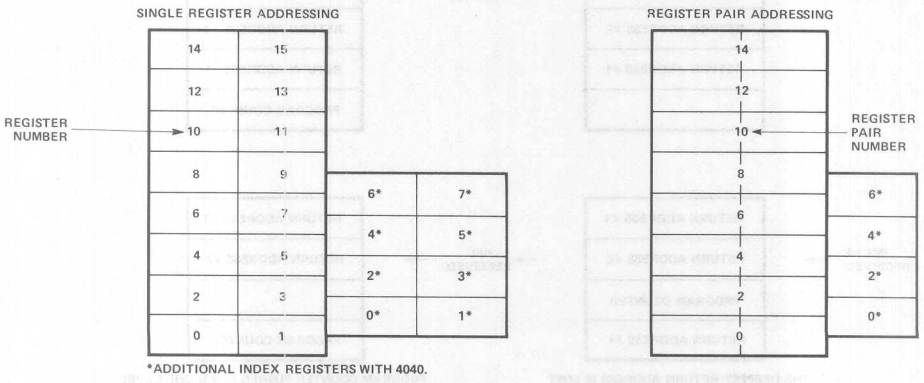


Figure 1-16. Index Register Organization.

last address is sent out. This new address then becomes the effective address. If a JMS (Jump to Subroutine) instruction is received by the CPU, the program control is transferred to the address called out in JMS instruction. This address is stored in the register just above the old program counter which now saves the address of the next instruction to be executed following the last JMS (1). This return address becomes the effective address following the BBL (Branch back and load) instruction at the end of the subroutine. The 4040 CPU can in addition to executing JMS/BBL combination to get to and from a subroutine, can also execute an Interrupt/BBS. In this case, the interrupt forces the program to location 003. The BBS returns the program counter to its previous value plus 1 and sends out the value stored in the SRC Save Register to restore port selection logic to the state that existed before the interrupt.

In summary, then, a JMS instruction pushes the program counter up one level and a BBL instruction pushes the program counter down one level. Since there are 7 registers in the push down stack of the 4040, 7 return addresses may be saved. If an eighth JMS occurs, the deepest return address (the first one stored) is lost. For a 4004, a fourth JMS occurrence without an intermediate return causes the loss of one return address. (see Figure 1-17 below).

Note 1: Since the JMS instruction is a 2-word instruction the old effective address is incremented by 2 to correctly give the address of the next instruction to be executed after the return from JMS.

Operation of the Command Lines and the SRC Command

The CPU command lines (CM-ROM, CM-RAM$_i$) are used to control the ROM's and RAM's by indicating how to interpret the data bus content at any given time.

The command lines allow the implementation of RAM bank, chip, register and character addressing, ROM chip addressing, as well as activating the instruction control in each ROM and RAM chip at the time the CPU receives an I/O and RAM group instruction.

Each CM-RAM$_i$ line can be selected by the execution of the DCL (Designate Command Line) instruction. The CM-ROM line, however, is always enabled for a 4004.

For a 4040, the CM-ROM$_0$ or CM-ROM$_1$ line is selected depending on the execution history of Designate ROM Bank instructions.

Note: If the number of ROMs in the system needs to be more than 16 for a 4004 or 32 for a 4040, external circuitry can be used to route CM-ROM to additional ROM banks. The same comment applies to the CM-RAM$_i$ lines if more than 16 RAMs need to be used.

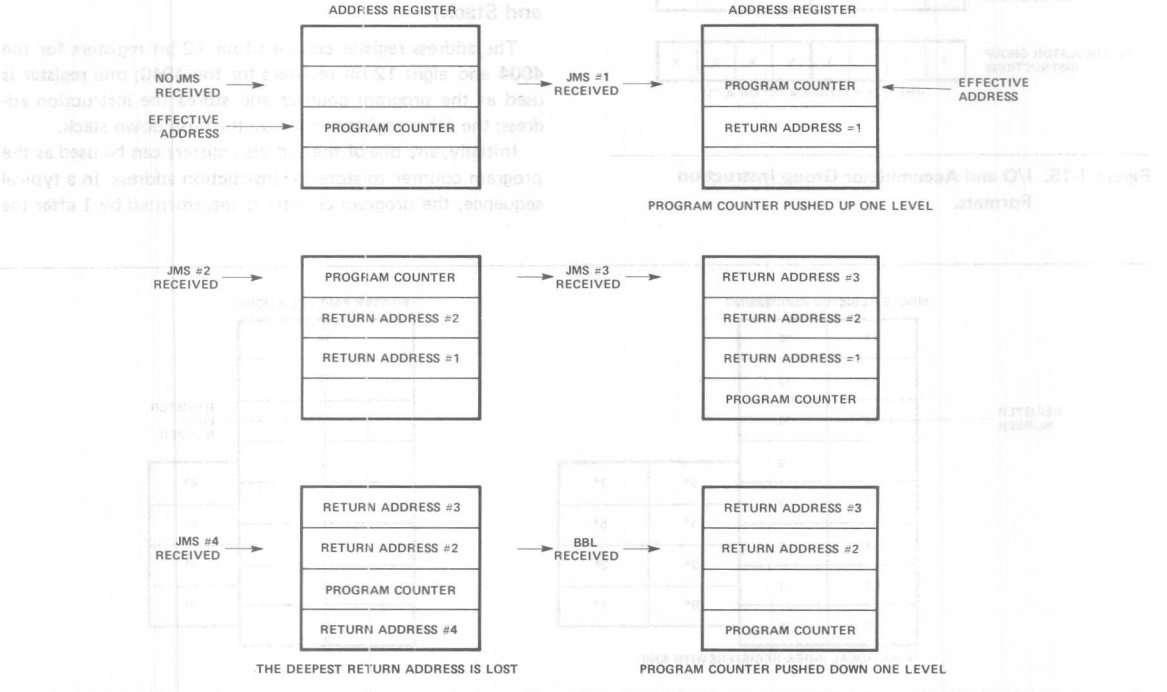


Figure 1-17. Operation of the Address Register on a Jump to Subroutine Instruction for 4004.

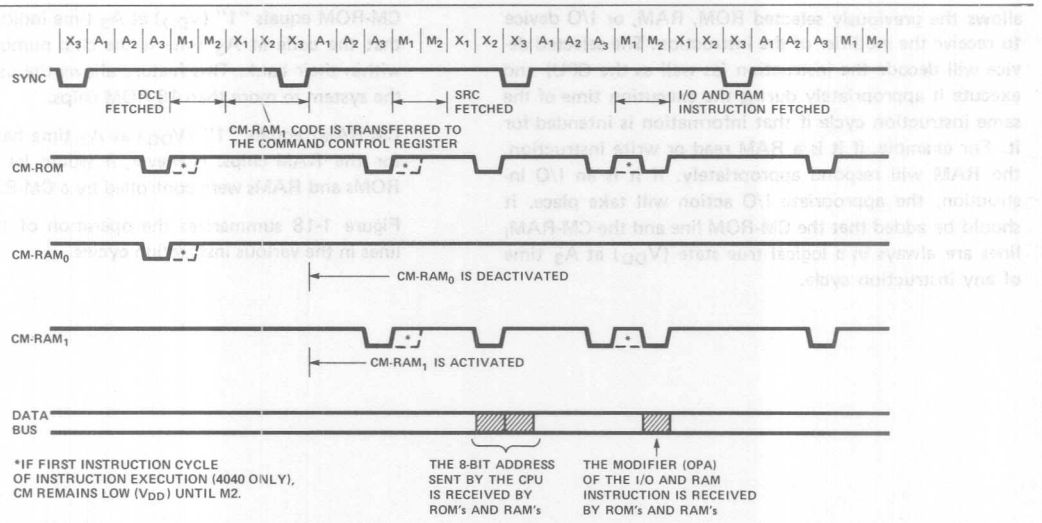


Figure 1-18. Operation of the Command Control Lines.

Execution of I/O and RAM Instructions

For the execution of an I/O and RAM group instruction the following steps are necessary:

1. The appropriate command line must be selected (by DCL).
2. The I/O port or RAM chip, register and character must be selected using the SRC (Send Register Control) instruction.
3. An I/O or RAM instruction must be fetched and executed (WRM, RDM, WRR, . . .).

The following is a detailed explanation of each step.

1. Prior to execution of the DCL instruction the desired CM-RAM$_i$ code must be stored in the accumulator (for example through an LDM instruction).

2. During DCL the CM-RAM$_i$ code is transferred from the accumulator to the command control register in the CPU. One CM-RAM$_i$ line is then activated (selecting one RAM bank) during the next instruction which would be an SRC. The CM-RAM$_i$ code remains in the command control register until a new DCL instruction is received. Each time a new SRC instruction is executed it will operate on the same RAM bank. This allows all RAM and I/O instructions to be executed on the same CM-RAM line without the necessity of executing another DCL instruction each time. DCL does not affect CM-ROM lines. Only the RAM or I/O device on the designated command line will latch the SRC.

   Note: If up to 4 RAM chips are used in a system, it is convenient to arrange them in a bank controlled by CM-RAM$_0$. This is because CM-RAM$_0$ is automatically selected after the application of at least one RESET (usually at start-up time). In this case DCL is unnecessary and Step 1 and 2 are omitted.

3. The SRC instruction specifies an index register pair in the CPU, whose content is an 8-bit address (this 8-bit address has previously been stored in the register pair). The 8-bit address is used to select one of the following: a) a RAM chip, register and character; b) a ROM chip; or c) an I/O device such as a 4269 or 4265 (and possibly a register or character within that device). The address is sent to the data bus during X_2 and X_3 time of the SRC instruction cycle. At X_2 time the selected CM-ROM line and the selected CM-RAM$_i$ line are in a logic true state (V_{DD}) to indicate which bank of RAMs, ROMs, and/or I/O devices are to respond to the 8-bit address that is now on the data bus. The 8-bit address is interpreted in the following way:

   by the ROMs:

   a. The first 4-bits (X_2 time) select one chip out of 16; a flip-flop is set in the selected chip.
   b. The second 4-bits (X_3 time) are ignored.

   by the RAMs:

   a. The first four bits sent out at X_2 time select one out of four chips and one out of four registers. The two higher order bits (D_3, D_2) select the chip and the two lower order bits (D_1, D_0) select the register.
   b. The second 4-bits (X_3 time) select one 4-bit character out of 16. The address is stored in the address register of the selected chip.

      (See Section 5 for a detailed description of the RAM chip.)

   by a 4265 Programmable I/O Device or a 4269 Programmable Keyboard/Display Device:

   a. See detailed description of components in Chapter 5.

4. After the SRC instruction has been executed, one ROM chip, I/O device or one RAM chip, register and character, have been selected. If the CPU fetches an I/O and RAM instruction, it will cause the CM-ROM and the selected CM-RAM$_i$ line to be logical true (V_{DD}) at M_2 time. This

allows the previously selected ROM, RAM, or I/O device to receive the modifier of the instruction. The selected device will decode the instruction (as well as the CPU) and execute it appropriately during the execution time of the same instruction cycle if that information is intended for it. For example, if it is a RAM read or write instruction, the RAM will respond appropriately. If it is an I/O instruction, the appropriate I/O action will take place. It should be added that the CM-ROM line and the CM-RAM$_i$ lines are always in a logical true state (V_{DD}) at A_3 time of any instruction cycle.

CM-ROM equals "1" (V_{DD}) at A_3 time indicates to ROMs that the code at A_3 time is the chip number of a ROM within their bank. This feature allows the user to expand the system to more than 16 ROM chips.

CM-RAM$_i$ equals "1" (V_{DD}) at A_3 time has no meaning for the RAM chips. However, it would be meaningful if ROMs and RAMs were controlled by a CM-RAM$_i$ line.

Figure 1-18 summarizes the operation of the command lines in the various instruction cycles.

INSTRUCTION SET SUMMARY

The instruction set of the 4040 and 4004 (CPU) are shown below. The following section will describe each instruction in detail.

[Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM]

MACHINE INSTRUCTIONS (Logic 1 = Low Voltage = Negative Voltage (V_{DD}); Logic 0 = High Voltage = (V_{SS})

MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
NOP	0 0 0 0	0 0 0 0	No operation.
*JCN	0 0 0 1 $A_2 A_2 A_2 A_2$	$C_1 C_2 C_3 C_4$ $A_1 A_1 A_1 A_1$	Jump to ROM address $A_2 A_2 A_2 A_2$, $A_1 A_1 A_1 A_1$ (within the same ROM that contains this JCN instruction) if condition $C_1 C_2 C_3 C_4$ [1] is true, otherwise skip (go to the next instruction in sequence).
*FIM	0 0 1 0 $D_2 D_2 D_2 D_2$	R R R 0 $D_1 D_1 D_1 D_1$	Fetch immediate (direct) from ROM Data D_2, D_1 to index register pair location RRR.[2]
SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X_2 and X_3 time in the Instruction Cycle.
FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A_1 and A_2 time in the Instruction Cycle.
*JUN	0 1 0 0 $A_2 A_2 A_2 A_2$	$A_3 A_3 A_3 A_3$ $A_1 A_1 A_1 A_1$	Jump unconditional to ROM address A_3, A_2, A_1.
*JMS	0 1 0 1 $A_2 A_2 A_2 A_2$	$A_3 A_3 A_3 A_3$ $A_1 A_1 A_1 A_1$	Jump to subroutine ROM address A_3, A_2, A_1, save old address. (Up 1 level in stack.)
INC	0 1 1 0	R R R R	Increment contents of register RRRR. [3]
*ISZ	0 1 1 1 $A_2 A_2 A_2 A_2$	R R R R $A_1 A_1 A_1 A_1$	Increment contents of register RRRR. Go to ROM address A_2, A_1 (within the same ROM that contains this ISZ instruction) if result $\neq 0$, otherwise skip (go to the next instruction in sequence).
ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with carry.
SUB	1 0 0 1	R R R R	Subtract contents of register RRRR to accumulator with borrow.
LD	1 0 1 0	R R R R	Load contents of register RRRR to accumulator.
XCH	1 0 1 1	R R R R	Exchange contents of index register RRRR and accumulator.
BBL	1 1 0 0	D D D D	Branch back (down 1 level in stack) and load data DDDD to accumulator.
LDM	1 1 0 1	D D D D	Load data DDDD to accumulator.

ACCUMULATOR GROUP INSTRUCTIONS

CLB	1 1 1 1	0 0 0 0	Clear both. (Accumulator and carry)
CLC	1 1 1 1	0 0 0 1	Clear carry.
IAC	1 1 1 1	0 0 1 0	Increment accumulator.
CMC	1 1 1 1	0 0 1 1	Complement carry.
CMA	1 1 1 1	0 1 0 0	Complement accumulator.
RAL	1 1 1 1	0 1 0 1	Rotate left. (Accumulator and carry)
RAR	1 1 1 1	0 1 1 0	Rotate right. (Accumulator and carry)
TCC	1 1 1 1	0 1 1 1	Transmit carry to accumulator and clear carry.
DAC	1 1 1 1	1 0 0 0	Decrement accumulator.
TCS	1 1 1 1	1 0 0 1	Transfer carry subtract and clear carry.
STC	1 1 1 1	1 0 1 0	Set carry.
DAA	1 1 1 1	1 0 1 1	Decimal adjust accumulator.
KBP	1 1 1 1	1 1 0 0	Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.
DCL	1 1 1 1	1 1 0 1	Designate command line.

INPUT/OUTPUT AND RAM INSTRUCTIONS for the following devices: 4001, 4002, 4008, 4009, and 4289*

(The RAM's and ROM's operated on in the I/O and RAM instructions have been previously selected by the last SRC instruction executed.)

MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
WRM	1 1 1 0	0 0 0 0	Write the contents of the accumulator into the previously selected RAM main memory character.
WMP	1 1 1 0	0 0 0 1	Write the contents of the accumulator into the previously selected RAM output port.
WRR	1 1 1 0	0 0 1 0	Write the contents of the accumulator into the previously selected ROM output port. (I/O Lines)
WPM	1 1 1 0	0 0 1 1	Write the contents of the accumulator into the previously selected half byte of read/write program memory (for use with 4008/4009 or 4289)
WRϕ (4)	1 1 1 0	0 1 0 0	Write the contents of the accumulator into the previously selected RAM status character 0.
WR1 (4)	1 1 1 0	0 1 0 1	Write the contents of the accumulator into the previously selected RAM status character 1.
WR2 (4)	1 1 1 0	0 1 1 0	Write the contents of the accumulator into the previously selected RAM status character 2.
WR3 (4)	1 1 1 0	0 1 1 1	Write the contents of the accumulator into the previously selected RAM status character 3.
SBM	1 1 1 0	1 0 0 0	Subtract the previously selected RAM main memory character from accumulator with borrow.
RDM	1 1 1 0	1 0 0 1	Read the previously selected RAM main memory character into the accumulator.
RDR	1 1 1 0	1 0 1 0	Read the contents of the previously selected ROM input port into the accumulator. (I/O Lines)
ADM	1 1 1 0	1 0 1 1	Add the previously selected RAM main memory character to accumulator with carry.
RDϕ (4)	1 1 1 0	1 1 0 0	Read the previously selected RAM status character 0 into accumulator.
RD1 (4)	1 1 1 0	1 1 0 1	Read the previously selected RAM status character 1 into accumulator.
RD2 (4)	1 1 1 0	1 1 1 0	Read the previously selected RAM status character 2 into accumulator.
RD3 (4)	1 1 1 0	1 1 1 1	Read the previously selected RAM status character 3 into accumulator.

*For explanation of 4265 and 4269 I/O instructions, see the 4265 and 4269 data sheets.

4040 ONLY INSTRUCTIONS

MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
HLT	0 0 0 0	0 0 0 1	Halt – inhibit program counter and data buffers.
BBS	0 0 0 0	0 0 1 0	Branch Back from Interrupt and restore the previous SRC. The Program Counter and send register control are restored to their pre-interrupt value.
LCR	0 0 0 0	0 0 1 1	The contents of the COMMAND REGISTER are transferred to the ACCUMULATOR.
OR4	0 0 0 0	0 1 0 0	The 4 bit contents of register #4 are logically "OR-ed" with the ACCUM.
OR5	0 0 0 0	0 1 0 1	The 4 bit contents of index register #5 are logically "OR-ed" with the ACCUMULATOR.
AN6	0 0 0 0	0 1 1 0	The 4 bit contents of index register #6 are logically "AND-ed" with the ACCUMULATOR
AN7	0 0 0 0	0 1 1 1	The 4 bit contents of index register #7 are logically "AND-ed" with the ACCUMULATOR.
DB0	0 0 0 0	1 0 0 0	DESIGNATE ROM BANK 0. CM-ROM$_0$ becomes enabled.
DB1	0 0 0 0	1 0 0 1	DESIGNATE ROM BANK 1. CM-ROM$_1$ becomes enabled.
SB0	0 0 0 0	1 0 1 0	SELECT INDEX REGISTER BANK 0. The index registers 0 - 7.
SB1	0 0 0 0	1 0 1 1	SELECT INDEX REGISTER BANK 1. The index registers 0* - 7*.
EIN	0 0 0 0	1 1 0 0	ENABLE INTERRUPT.
DIN	0 0 0 0	1 1 0 1	DISABLE INTERRUPT.
RPM	0 0 0 0	1 1 1 0	READ PROGRAM MEMORY.

NOTES: (1) The condition code is assigned as follows:

$C_1 = 1$ Invert jump condition $C_2 = 1$ Jump if accumulator is zero $C_4 = 1$ Jump if test signal is a 0

$C_1 = 0$ Not invert jump condition $C_3 = 1$ Jump if carry/link is a 1

(2) RRR is the address of 1 of 8 index register pairs in the CPU.

(3) RRRR is the address of 1 of 16 index registers in the CPU.

(4) Each RAM chip has 4 registers, each with twenty 4-bit characters subdivided into 16 main memory characters and 4 status characters. Chip number, RAM register and main memory character are addressed by an SRC instruction. For the selected chip and register, however, status character locations are selected by the instruction code (OPA).

DETAILED INSTRUCTION DESCRIPTION

A. Symbols and Abbreviations

The following symbols and abbreviations will be used throughout the next few sections:

SRCR	SRC Register
( )	the content of
⟶	is transferred to
ACC	Accumulator (4 bit)
CY	Carry Flip-Flop
ACBR	Accumulator Buffer Register (4 bit)
RRRR	Index register address
RRR	Index register pair address
P_L	Low order program counter Field (4 bit)
P_M	Middle order program counter Field (4 bit)
P_H	High order program counter Field (4 bit)
a_i	Order i content of the accumulator
CM_i	Order i content of the command register
M	RAM main character location
M_{si}	RAM status character i
DB (T)	Data bus content at time T
Stack	The 3 or 7 registers in the address register other than the program counter.
CR	Command register
IE	Interrupt enable
RB0	Register bank 0 $RRRR_0$ – RRRR, enable
RB1	Register bank 1 $RRRR_0$ – RRRR, enable
V	Logical OR
Λ	Logical AND

Throughout the text "page" means a block of 256 instructions whose address differs only on the least significant 8 bits; e.g., page 7 means all locations having addresses between 0111 0000 0000 and 0111 1111 1111.

B. Format for Describing Each Instruction

Each instruction will be described as follows:

(1) Mnemonic symbol and meaning
(2) OPR and OPA code
(3) Symbolic representation of the instruction
(4) Description of the instruction (if necessary)
(5) Example and/or exceptions (if necessary)

C. One Word Machine Instructions

Mnemonic: **NOP** (No Operation)
OPR OPA: 0000 0000
Symbolic: Not applicable
Description: No operation performed

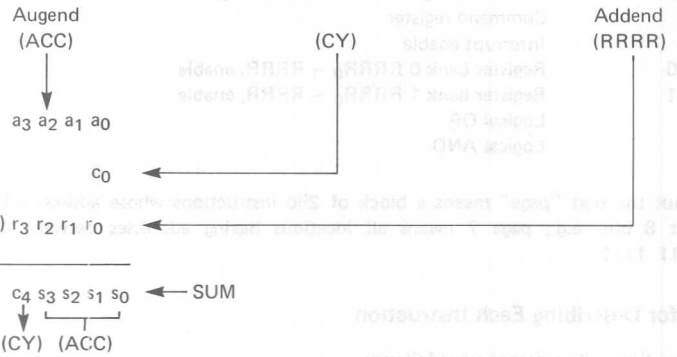
Mnemonic: **LDM** (Load Data to Accumulator)
OPR OPA: 1101 DDDD
Symbolic: DDDD → ACC
Description: The 4 bits of data, DDDD stored in the OPA field of instruction word are loaded into the accumulator. The previous contents of the accumulator are lost. The carry/link bit is unaffected.

Mnemonic: **LD** (Load index register to Accumulator)
OPR OPA: 1010 RRRR
Symbolic: (RRRR) → ACC
Description: The 4 bit content of the designated index register (RRRR) is loaded into the accumulator. The previous contents of the accumulator are lost. The 4 bit content of the index register and the carry/link bit are unaffected.

Mnemonic: **XCH** (Exchange index register and accumulator)
OPR OPA: 1011 RRRR
Symbolic: (ACC) → ACBR, (RRRR) → ACC, (ACBR) → RRRR
Description: The 4 bit content of the designated index register is loaded into the accumulator. The prior content of the accumulator is loaded into the designated register. The carry/link bit is unaffected.

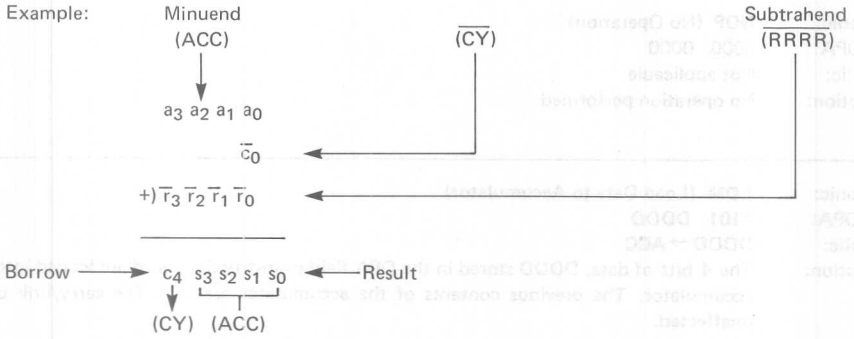
Mnemonic: **ADD** (Add index register to accumulator with carry)
OPR OPA: 1000 RRRR
Symbolic: (RRRR) + (ACC) + (CY) → ACC, CY
Description: The 4 bit content of the designated index register is added to the content of the accumulator with carry. The result is stored in the accumulator. The carry/link is set to 1 if a sum greater than 15_{10} was generated to indicate a carry out; otherwise, the carry/link is set to 0. The 4 bit content of the index register is unaffected.

Example:



Mnemonic: **SUB** (Subtract index register from accumulator with borrow)
OPR OPA: 1001 RRRR
Symbolic: (ACC) + $(\overline{RRRR})$ + $(\overline{CY})$ → ACC, CY
Description: The 4 bit content of the designated index register is complemented (ones complement) and added to content of the accumulator with borrow and the result is stored in the accumulator. If a borrow is generated, the carry bit is set to 0; otherwise, it is set to 1. The 4 bit content of the index register is unaffected.

Example:



Mnemonic: **INC** (Increment index register)
OPR OPA: 0110 RRRR
Symbolic: (RRRR) +1 → RRRR
Description: The 4 bit content of the designated index register is incremented by 1. The index register is set to zero in case of overflow. The carry/link is unaffected.

Mnemonic: **BBL** (Branch back and load data to the accumulator)
OPR OPA: 1100 DDDD
Symbolic: (Stack) → P_L, P_M, P_H; DDDD → ACC
Description: The program counter (address stack) is pushed down one level. Program control transfers to the next instruction following the last jump to subroutine (JMS) instruction. The 4 bits of data DDDD stored in the OPA portion of the instruction are loaded to the accumulator. BBL is used to return from subroutine to main program.

Mnemonic: **JIN** (Jump indirect)
OPR OPA: 0011 RRR1
Symbolic: (RRR0) → P_M
(RRR1) → P_L; P_H unchanged
Description: The 8 bit content of the designated index register pair is loaded into the low order 8 positions of the program counter. Program control is transferred to the instruction at that address on the same page (same ROM) where the JIN instruction is located. The 8 bit content of the index register is unaffected.

EXCEPTIONS: When JIN is located at the address (P_H) 1111 1111 program control is transferred to the next page in sequence and not to the same page where the JIN instruction is located. That is, the next address is (P_H + 1) (RRR0) (RRR1) and not (P_H) (RRR0) (RRR1)

Mnemonic: **SRC** (Send register control)
OPR OPA: 0010 RRR1
Symbolic: (RRR0) → DB (X_2)
(RRR1) → DB (X_3)
Description: The 8 bit content of the designated index register pair is sent to the RAM address register at X_2 and X_3. A subsequent read, write, or I/O operation of the RAM will utilize this address. Specifically, the first 2 bits of the address designate a RAM chip; the second 2 bits designate 1 out of 4 registers within the chip; the last 4 bits designate 1 out of 16 4 bit main memory characters within the register. This command is also used to designate a ROM I/O port or an I/O device for a subsequent I/O operation. The address in I/O device, ROM or RAM is not cleared until the next SRC instruction is executed. The 8 bit content of the index register is unaffected.

Mnemonic: **FIN** (Fetch indirect from ROM)
OPR OPA: 0011 RRR0
Symbolic: (P_H) (0000) (0001) → ROM address
(OPR) → RRR0
(OPA) → RRR1
Description: The 8 bit content of the 0 index register pair (0000) (0001) is sent out as an address in the same page where the FIN instruction is located. The 8 bit word at that location is loaded into the designated index register pair. The program counter is unaffected; after FIN has been executed the next instruction in sequence will be addressed. The content of the 0 index register pair is unaltered unless index register 0 was designated.

EXCEPTIONS:
a. Although FIN is a 1-word instruction, its execution requires two instruction cycles.
b. When FIN is located at address (P_H) 1111 1111 data will be fetched from the next page (ROM) in sequence and not from the same page (ROM) where the FIN instruction is located. That is, next address is (P_H + 1) (0000) (0001) and not (P_H) (0000) (0001).

Mnemonic: **HLT**
OPR OPA: 0000 0001
Symbolic: 1 → HALT 1 → STOP
Description: The processor sets the HALT and STOP flip-flops. Program counter incrementer and data input buffers are inhibited. The processor executes NOP continuously; continuation can occur by means of STOP or INTERRUPT control.

In this mode, the Program Counter + 1 is gated out at A_1, A_2, and A_3, times on the data bus. M_1, M_2 times will contain the addressed ROM instruction on the data bus. X_1 will contain the 4 bit Accumulator contents. X_2 and X_3 will contain the 8 bit SRC register.

Mnemonic: **BBS**
OPR OPA: 0000 0010
Symbolic: (Stack → P_L, P_M, P_H;)
SRCR0 → DB(X2)
SRCR1 → DB(X3)
Description: This instruction is a combination of BRANCH BACK and SRC. The effective address counter is decremented and program control is returned to the location saved by the forced JMS which occurred at the beginning of the interrupt routine. In addition, the content of the SRC register is sent out at X_2 and X_3 of the instruction cycle, thus restoring the I/O port selection. This instruction will also turn off the INTA line re-enabling the CPU for Interrupt.

The previously selected Index register bank will also be restored during this instruction.

Mnemonic: **LCR**
OPR OPA: 0000 0011
Symbolic: (CR) → ACC
Description: The 4 bit contents of the COMMAND REGISTER are transferred to the ACCUMULATOR. This allows saving the command register values before processing the interrupt.

Mnemonic: **OR4**
OPR OPA: 0000 0100
Symbolic: ($RRRR_4$) V (ACC) → ACC
Description: The 4 bit contents of index register #4 are logically "OR-ed" with the ACCUMULATOR. The result is placed in the ACCUMULATOR and the CARRY flip-flop is unaffected.

Examples:

(ACC)	0101		(ACC)	0000
($RRRR_4$)	1001		($RRRR_4$)	1000
ACC	1101		ACC	1000

Mnemonic: **OR5**
OPR OPA: 0000 0101
Symbolic: ($RRRR_5$) V (ACC) → ACC
Description: The 4 bit contents of index register #5 are logically "OR-ed" with the ACCUMULATOR. Carry flip-flop is unaffected.

Mnemonic: **AN6**
OPR OPA: 0000 0110
Symbolic: ($RRRR_6$) Λ (ACC) → ACC
Description: The 4 bit contents of index register #6 are logically "AND-ed" with the ACCUMULATOR. The result is placed in the ACCUMULATOR and the CARRY is unaffected.

Examples:

(ACC)	0110		(ACC)	1111
($RRRR_6$)	0100		($RRRR_6$)	0001
ACC	0100		ACC	0001

Mnemonic: **AN7**
OPR OPA: 0000 0111
Symbolic: $(RRRR_7) \wedge (ACC) \rightarrow ACC$
Description: The 4 bit contents of index register #7 are logically "AND-ed" with the ACCUMULATOR. Carry flip-flop is unaffected.

Mnemonic: **DB0**
OPR OPA: 0000 1000
Symbolic: Enable $\rightarrow$ CM-ROM_0
Description: DESIGNATE ROM BANK 0. The most significant bit of the COMMAND REGISTER, CR_3, is reset. On the third instruction cycle following its execution, it causes CM-ROM_0 to be activated. This Bank is selected with reset.

Mnemonic: **DB1**
OPR OPA: 0000 1001
Symbolic: Enable $\rightarrow$ CM-ROM_1
Description: DESIGNATE ROM BANK 1. The most significant bit of the COMMAND REGISTER, CR_3, is set. On the third instruction cycle following its execution, it causes CM-ROM_1 to be activated.

Mnemonic: **SB0**
OPR OPA: 0000 1010
Symbolic: $1 \rightarrow RB0$, $0 \rightarrow RB1$
Description: SELECT INDEX REGISTER BANK 0. The index register bank select flip-flop is reset. Index registers 0 - 7, 8 - 15 will be available for program use. This bank is to be selected with a Reset.

Mnemonic: **SB1**
OPR OPA: 0000 1011
Symbolic: $0 \rightarrow RB0$ $1 \rightarrow RB1$
Description: SELECT INDEX REGISTER BANK 1. The index register bank select flip-flop is set. Index registers 0^* - 7^*, 8 - 15 will be available for program use.

Mnemonic: **WPM**
OPR OPA: 1110 0011
Symbolic: (1111) (SRC) $\rightarrow$ ROM/RAM address bus (4289)
(ACC) $\rightarrow$ I/O_0-I/O_3 (4289)
Description: WRITE PROGRAM MEMORY. When an instruction is to be stored in RAM program memory, it is written in two four-bit segments. The F/L signal from the 4289 keeps track of which half is being written. When the CPU executes a WPM instruction, the chip select lines of the 4289 are jammed with "1111". In the system design this should be designated as the RAM channel. The OUT and PM line on the 4289 are also activated by the WPM instruction. The previously selected SRC address on line A0 through A7 of the 4289 becomes the address of the RAM word being written. By appropriately decoding the chip select lines, the W line, and F/L, the write strobes can be generated for the memory. The F/L line is initially at logic 0 (V_{SS}) when power comes on and after reset. It then pulses to logic 1 (V_{DD}) when every second WPM is executed. A high on the F/L line means that the first four bits are being written, and a low means that the last four bits are being written. The 4289 transfers the segment of the instruction to the I/O bus at X2 of the WPM instruction. The SRC address sent to RAM is only 8-bits. When more than one page of RAM (256 bytes) is being written, an output port must be used to supply additional address lines for higher order addresses.

Mnemonic: **RPM**
OPR OPA: 0000 1110
Symbolic: (1111) (SRC) → ROM/RAM address bus (4289)
(DDDD) → ACC
Description: READ PROGRAM MEMORY. This instruction can be used only with the 4289 Standard Memory Chip. The contents of the previously selected nibble of R/W Program Memory are transferred to the 4040 and loaded to the ACCUMULATOR. The F/L 4289 control line is toggled with each RPM. The 4289 IN line and PM line are also active during this instruction.

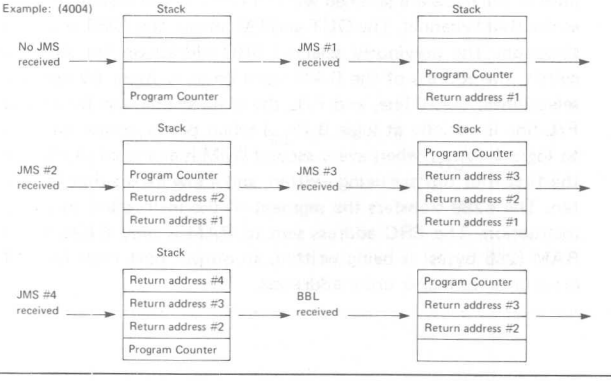
Mnemonic: **EIN**
OPR OPA: 0000 1100
Symbolic: 1 → IE
Description: ENABLE INTERRUPT. Internal interrupt detection logic is enabled.

Mnemonic: **DIN**
OPR OPA: 0000 1101
Symbolic: 0 → IE
Description: DISABLE INTERRUPT. Internal interrupt detection logic is disabled.

D. Two Word Machine Instruction

Mnemonic: **JUN** (Jump unconditional)
1st word OPR OPA: 0100 A_3 A_3 A_3 A_3
2nd word OPR OPA: A_2 A_2 A_2 A_2 A_1 A_1 A_1 A_1
Symbolic: A_1 A_1 A_1 A_1 → P_L, A_2 A_2 A_2 A_2 → P_M, A_3 A_3 A_3 A_3 → P_H
Description: Program control is unconditionally transferred to the instruction locator at the address A_3 A_3 A_3 A_3, A_2 A_2 A_2 A_2, A_1 A_1 A_1 A_1

Mnemonic: **JMS** (Jump to Subroutine)
1st word OPR OPA: 0101 A_3 A_3 A_3 A_3
2nd word OPR OPA: A_2 A_2 A_2 A_2 A_1 A_1 A_1
Symbolic: (P_H, P_M, P_L + 2) → Stack
A_1 A_1 A_1 A_1 → P_L, A_2 A_2 A_2 A_2 → P_M, A_3 A_3 A_3 A_3 → P_H
Description: The address of the next instruction in sequence following JMS (return address) is saved in the push down stack. Program control is transferred to the instruction located at the 12 bit address ($A_3A_3A_3A_3A_2A_2A_2A_2A_1A_1A_1A_1$). Execution of a return instruction (BBL) will cause the saved address to be pulled out of the stack, therefore, program control is transferred to the next sequential instruction after the last JMS.
The push down stack has 4 registers (8 registers in 4040). One of them is used as the program counter, therefore nesting of JMS can occur up to 3 levels (7 levels in the 4040).



Mnemonic: **JCN** (Jump conditional)
1st word OPR OPA: 0001 $C_1\ C_2\ C_3\ C_4$
2nd word OPR OPA: $A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1\ A_1$
Symbolic:
If $C_1\ C_2\ C_3\ C_4$ is true, $A_2\ A_2\ A_2\ A_2 \rightarrow P_M$
$A_1\ A_1\ A_1\ A_1 \rightarrow P_L$, P_H unchanged
if $C_1\ C_2\ C_3\ C_4$ is false,
$(P_H) \rightarrow P_H$, $(P_M) \rightarrow P_M$, $(P_L + 2) \rightarrow P_L$

Description: If the designated condition code is true, program control is transferred to the instruction located at the 8 bit address $A_2\ A_2\ A_2\ A_2$, $A_1\ A_1\ A_1\ A_1$ on the same page (ROM) where JCN is located.

If the condition is not true the next instruction in sequence after JCN is executed.

The condition bits are assigned as follows:

$C_1 = 0$ Do not invert jump condition
$C_1 = 1$ Invert jump condition
$C_2 = 1$ Jump if the accumulator content is zero
$C_3 = 1$ Jump if the carry/link content is 1
$C_4 = 1$ Jump if test signal (pin 10 on 4004) is zero.

C_X Condition Table for JCN Instruction

C_1	C_2	C_3	C_4	
0	0	0	0	NO OPERATION
0	0	0	1	Jump if test = 0 (V_{SS})
0	0	1	0	Jump if CY = 1
0	0	1	1	Jump if test = 0 or CY = 1
0	1	0	0	Jump if AC = 0
0	1	0	1	Jump if test = 0 or AC = 0
0	1	1	0	Jump if CY = 1 or AC = 0
0	1	1	1	Jump if test = 0 or CY = 1 or AC = 0
1	0	0	0	Jump Unconditionally
1	0	0	1	Jump if test = 1 (V_{DD})
1	0	1	0	Jump if CY = 0
1	0	1	1	Jump if test = 1 and CY = 0
1	1	0	0	Jump if AC $\neq$ 0
1	1	0	1	Jump if test = 1 and AC $\neq$ 0
1	1	1	0	Jump if CY = 0 and AC $\neq$ 0
1	1	1	1	Jump if test = 1 and CY = 0 and AC $\neq$ 0

NOTE: A logic "1" is the most negative test input. A logic "0" is the most positive test input.

Example:

OPR	OPA	
0001	0110	Jump if accumulator is zero or carry = 1

Several conditions can be tested simultaneously.

The logic equation describing the condition for a jump is given below:

$$JUMP = \overline{C_1} \cdot ((ACC = 0) \cdot C_2 + (CY = 1) \cdot C_3 + \overline{TEST} \cdot C_4) +$$

$$C_1 \cdot \overline{((ACC = 0) \cdot C_2 + (CY = 1) \cdot C_3 + \overline{TEST} \cdot C_4)}$$

EXCEPTIONS: If JCN is located on words 254 and 255 of a ROM page, when JCN is executed and the condition is true, program control is transferred to the 8-bit address on the page following where JCN is located.

Mnemonic: **ISZ** (Increment index register skip if zero)
1st word OPR OPA: 0111 RRRR
2nd word OPR OPA: $A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1\ A_1$
Symbolic: (RRRR) + 1 → RRRR, if result = 0
$(P_H) \rightarrow P_H$, $(P_M) \rightarrow P_M$, $(P_L + 2) \rightarrow P_L$:
if result ≠ 0 $(P_H) \rightarrow P_H$,
$A_2\ A_2\ A_2\ A_2 \rightarrow P_M$, $A_1\ A_1\ A_1\ A_1 \rightarrow P_L$
Description: The content of the designated index register is incremented by 1. The accumulator and carry/link are unaffected. If the result is zero, the next instruction after ISZ is executed. If the result is different from 0, program control is transferred to the instruction located at the 8 bit address $A_2\ A_2\ A_2\ A_2$, $A_1\ A_1\ A_1\ A_1$ on the same page (ROM) where the ISZ instruction is located.

EXCEPTIONS: If ISZ is located on words 254 and 255 of a ROM page, when ISZ is executed and the result is not zero, program control is transferred to the 8 bit address located on the next page in sequence and not on the same page where ISZ is located.

Mnemonic: **FIM** (Fetched immediate from ROM)
1st word OPR OPA: 0010 RRR0
2nd word OPR OPA: $D_2\ D_2\ D_2\ D_2$ $D_1\ D_1\ D_1\ D_1$
Symbolic: $D_2\ D_2\ D_2\ D_2 \rightarrow$ RRR0
$D_1\ D_1\ D_1\ D_1 \rightarrow$ RRR1
Description: The 2nd word represents 8 bits of data which are loaded into the designated index register pair.

E. Input/Output Instructions

The following I/O instructions are described as they relate to ROM and RAM devices. These same instructions (mnemonics) can be redefined for devices other than ROM and RAM.

Mnemonic: **RDM** (Read RAM character)
OPR OPA: 1110 1001
Symbolic: (M) → ACC
Description: The content of the previously selected RAM main memory character is transferred to the accumulator. The carry/link is unaffected. The 4 bit data in memory is unaffected.

Mnemonic: **RDO** (Read RAM status character 0)
OPR OPA: 1110 1100
Symbolic: $(M_{S0}) \rightarrow$ ACC
Description: The 4 bits of status character 0 for the previously selected RAM register are transferred to the accumulator. The carry/link and the status character are unaffected.

Mnemonic: **RD1** (Read RAM status character 1)
OPR OPA: 1110 1101
Symbolic: $(M_{S1}) \rightarrow$ ACC

Mnemonic: **RD2** (Read RAM status character 2)
OPR OPA: 1110 1110
Symbolic: $(M_{S2}) \rightarrow$ ACC

Mnemonic: **RD3** (Read RAM status character 3)
OPR OPA: 1110 1111
Symbolic: $(M_{S3}) \rightarrow$ ACC

Mnemonic: **RDR** (Read ROM port)
OPR OPA: 1110 1010
Symbolic: (ROM input lines) → ACC
Description: The data present at the input lines of the previously selected ROM chip is transferred to the accumulator. The carry/link is unaffected.

If the I/O option has both inputs and outputs within the same 4 I/O lines, the user can choose to have either "0" or "1" transferred to the accumulator for those I/O pins coded as outputs, when an RDR instruction is executed.

Example: Given a port with I/O coded with 2 inputs and 2 outputs, when RDR is executed the transfer is as shown below:

$I_3\ O_2\ O_1\ I_0$ (ACC)

1 X X 0 ⟹ 1 (1 or 0) (1 or 0) 0

Input Data — User can choose

Mnemonic: **WRM** (Write accumulator into RAM character)
OPR OPA: 1110 0000
Symbolic: (ACC) → M
Description: The accumulator content is written into the previously selected RAM main memory character location. The accumulator and carry/link are unaffected.

Mnemonic: **WRO** (Write accumulator into RAM status character 0)
OPR OPA: 1110 0100
Symbolic: (ACC) → M_{S0}
Description: The content of the accumulator is written into the RAM status character 0 of the previously selected RAM register. The accumulator and the carry/link are unaffected.

Mnemonic: **WR1** (Write accumulator into RAM status character 1)
OPR OPA: 1110 0101
Symbolic: (ACC) → M_{S1}

Mnemonic: **WR2** (Write accumulator into RAM status character 2)
OPR OPA: 1110 0110
Symbolic: (ACC) → M_{S2}

Mnemonic: **WR3** (Write accumulator into RAM status character 3)
OPR OPA: 1110 0111
Symbolic: (ACC) → M_{S3}

Mnemonic: **WRR** (Write ROM port)
OPR OPA: 1110 0010
Symbolic: (ACC) → ROM output lines
Description: The content of the accumulator is transferred to the ROM output port of the previously selected ROM chip. The data is available on the output pins until a new WRR is executed on the same chip. The ACC content and carry/link are unaffected. (The LSB bit of the accumulator appears on I/O_0.) No operation is performed on I/O lines coded as inputs.

Mnemonic: **WMP** (Write memory port)
OPR OPA: 1110 0001
Symbolic: (ACC) → RAM output register
Description: The content of the accumulator is transferred to the RAM output port of the previously selected RAM chip. The data is available on the output pins until a new WMP is executed on the same RAM chip. The content of the ACC and the carry/link are unaffected. (The LSB bit of the accumulator appears on O_0, Pin 16, of the 4002.)

Mnemonic: **ADM** (Add from memory with carry)
OPR OPA: 1110 1011
Symbolic: (M) + (ACC) + (CY) → ACC, CY
Description: The content of the previously selected RAM main memory character is added to the accumulator with carry. The RAM character is unaffected.

Mnemonic: **SBM** (Subtract from memory with borrow)
OPR OPA: 1110 1000
Symbolic: $(\overline{M}) + (ACC) + (\overline{CY}) \rightarrow ACC, CY$
Description: The content of the previously selected RAM character is subtracted from the accumulator with borrow. The RAM character is unaffected.

F. Accumulator Group Instructions

Mnemonic: **CLB** (Clear both)
OPR OPA: 1111 0000
Symbolic: 0 → ACC, 0 → CY
Description: Set accumulator and carry/link to 0.

Mnemonic: **CLC** (Clear carry)
OPR OPA: 1111 0001
Symbolic: 0 → CY
Description: Set carry/link to 0

Mnemonic: **CMC** (Complement carry)
OPR OPA: 1111 0011
Symbolic: $(\overline{CY}) \rightarrow CY$
Description: The carry/link content is complemented

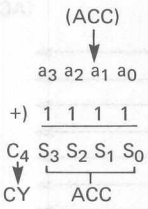
Mnemonic: **STC** (Set carry)
OPR OPA: 1111 1010
Symbolic: 1 → CY
Description: Set carry/link to a 1

Mnemonic: **CMA** (Complement Accumulator)
OPR OPA: 1111 0100
Symbolic: $\overline{a_3}\ \overline{a_2}\ \overline{a_1}\ \overline{a_0} \rightarrow ACC$
Description: The content of the accumulator is complemented. The carry/link is unaffected.

Mnemonic: **IAC** (Increment accumulator)
OPR OPA: 1111 0010
Symbolic: (ACC) + 1 → ACC
Description: The content of the accumulator is incremented by 1. No overflow sets the carry/link to 0; overflow sets the carry/link to a 1.

Mnemonic: **DAC** (decrement accumulator)
OPR OPA: 1111 1000
Symbolic: (ACC) − 1 → ACC
Description: The content of the accumulator is decremented by 1. A borrow sets the carry/link to 0, no borrow sets the carry/link to a 1.

Example:



Mnemonic: **RAL** (Rotate left)
OPR OPA: 1111 0101
Symbolic: $C_0 \rightarrow a_0$, $a_i \rightarrow a_{i+1}$, $a_3 \rightarrow CY$
Description: The content of the accumulator and carry/link are rotated left.

Mnemonic: **RAR** (Rotate right)
OPR OPA: 1111 0110
Symbolic: $a_0 \rightarrow CY$, $a_i \rightarrow a_{i-1}$, $C_0 \rightarrow a_3$
Description: The content of the accumulator and carry/link are rotated right.

Mnemonic: **TCC** (Transmit carry and clear)
OPR OPA: 1111 0111
Symbolic: $0 \rightarrow ACC$, $(CY) \rightarrow a_0$, $0 \rightarrow CY$
Description: The accumulator is cleared. The least significant position of the accumulator is set to the value of the carry/link. The carry/link is set to 0.

Mnemonic: **DAA** (Decimal adjust accumulator)
OPR OPA: 1111 1011
Symbolic: (ACC) + 0000 → ACC
or
0110
Description: The accumulator is incremented by 6 if either the carry/link is 1 or if the accumulator content is greater than 9. The carry/link is set to a 1 if the result generates a carry, otherwise it is unaffected.

Mnemonic: **TCS** (Transfer carry subtract)
OPR OPA: 1111 1001
Symbolic: 1001 → ACC if (CY) = 0
1010 → ACC if (CY) = 1
0 → CY
Description: The accumulator is set to 9 if the carry/link is 0.
The accumulator is set to 10 if the carry/link is a 1.
The carry/link is set to 0.

Mnemonic: **KBP** (Keyboard process)
OPR OPA: 1111 1100
Symbolic: (ACC) → KBP ROM → ACC
Description: A code conversion is performed on the accumulator content, from 1 out of n to binary code. If the accumulator content has more than one bit on, the accumulator will be set to 15 (to indicate error). The carry/link is unaffected. The conversion table is shown below.

(ACC) before KBP		(ACC) after KBP
0 0 0 0	→	0 0 0 0
0 0 0 1	→	0 0 0 1
0 0 1 0	→	0 0 1 0
0 1 0 0	→	0 0 1 1
1 0 0 0	→	0 1 0 0
0 0 1 1	→	1 1 1 1
0 1 0 1	→	1 1 1 1
0 1 1 0	→	1 1 1 1
0 1 1 1	→	1 1 1 1
1 0 0 1	→	1 1 1 1
1 0 1 0	→	1 1 1 1
1 0 1 1	→	1 1 1 1
1 1 0 0	→	1 1 1 1
1 1 0 1	→	1 1 1 1
1 1 1 0	→	1 1 1 1
1 1 1 1	→	1 1 1 1

Mnemonic: **DCL** (Designate command line)
OPR OPA: 1111 1101
Symbolic: $a_0 \rightarrow CM_0$, $a_1 \rightarrow CM_1$, $a_2 \rightarrow CM_2$
Description: The content of the three least significant accumulator bits is transferred to the command control register within the CPU.

This instruction provides RAM bank selection when multiple RAM banks are used. (If no DCL instruction is sent out, RAM Bank number zero is automatically selected after application of at least one RESET). DCL remains latched until it is changed.

The selection is made according to the following truth table.

(ACC)	CM - RAM_i Enabled	Bank No.
X 0 0 0	CM - RAM_0	Bank 0
X 0 0 1	CM - RAM_1	Bank 1
X 0 1 0	CM - RAM_2	Bank 2
X 1 0 0	CM - RAM_3	Bank 3
X 0 1 1	CM - RAM_1, CM - RAM_2	Bank 4
X 1 0 1	CM - RAM_1, CM - RAM_3	Bank 5
X 1 1 0	CM - RAM_2, CM - RAM_3	Bank 6
X 1 1 1	CM - RAM_1, CM - RAM_2, CM - RAM_3	Bank 7

INSTRUCTION	DATA @ X_2 $D_3 D_2 D_1 D_0$	DATA @ X_3 $D_3 D_2 D_1 D_0$	COMMENTS
NOP	1 1 1 1	1 1 1 1	
JCN	1 1 1 1	1 1 1 1	
A_2, A_1	1 1 1 1	1 1 1 1	
FIM RRR0	(RRR0)	(RRR1)	The content of address
$D_2 D_1$	1 1 1 1	1 1 1 1	pair RRR
SRC RRR1	(RRR0)	(RRR1)	
FIN RRR0	(RRR0)	(RRR1)	
2nd cycle	1 1 1 1	1 1 1 1	
JIN RRR0	(RRR0)	(RRR1)	
JUN A_3	A_3	1 1 1 1	
A_2, A_1	A_3	1 1 1 1	
JMS A_3	A_3	1 1 1 1	
A_2, A_1	A_3	1 1 1 1	
INC RRRR	(RRRR)	(RRRR) +1	Content of register RRRR;
ISZ RRRR	(RRRR)	(RRRR) +1	Content +1 of RRRR
A_2, A_1	1 1 1 1	1 1 1 1	
ADD RRRR	(RRRR)	1 1 1 1	Content of register RRR
SUB RRRR	(RRRR)	1 1 1 1	
LD RRRR	(RRRR)	1 1 1 1	
XCH RRRR	(RRRR)	(ACC)	Content of register RRRR; the content of ACC
BBL	DDDD	1 1 1 1	Data DDDD
LDM	DDDD	1 1 1 1	Data DDDD
WRM, WR0, WR1,	(ACC)	1 1 1 (CY) For 4004	Content of accumulator;
WR2, WR3,		1 1 1 1 For 4040	Content of CY F/F is
WPM, WMP, WRR	(ACC)	1 1 1 (CY) For 4004	present on D_0
		1 1 1 1 For 4040	
RDM, RD0, RD1, RD2 RD3, ADM, SBM, RDR	(M) or (Input)	(M) or (INPUT)	Data fetched from RAM or input
CLB, CLC, IAC, CMC CMA, RAL, PAR, TCC	0 0 0 0	1 1 1 1	
TCS	1 0 0 1	1 1 1 1	
STC, DAC, DCL	1 1 1 1	1 1 1 1	
DAA	0 0 0 0 or 0 1 1 0	1 1 1 1	X_2 depends on ACC content
KBP	0000,0001,0010 0011,0100,1111	1 1 1 1	X_2 depends on ACC content

Figure 1-19. MCS-40™ Data Bus Content During Execution of Each Instruction.

INSTRUCTION	DATA @ X_2 $D_3D_2D_1D_0$	DATA @ X_3 $D_3D_2D_1D_0$	COMMENTS
HLT	(SRCH)	(SRCL)	After execution of HLT, processor enters STOP mode.
BBS	(SRCH)	(SRCL)	(SRCH) means contents of 4 high order bits of SRC register.
LCR	(COM. REG)	1 1 1 1	
OR4	(0100)	1 1 1 1	
OR5	(0101)	1 1 1 1	
AN6	(0110)	1 1 1 1	
AN7	(0111)	1 1 1 1	
DB0	1 1 1 1	1 1 1 1	
DB1	1 1 1 1	1 1 1 1	
SB0	1 1 1 1	1 1 1 1	
SB1	1 1 1 1	1 1 1 1	
EIN	1 1 1 1	1 1 1 1	
DIN	1 1 1 1	1 1 1 1	
RPM	(P.M.)	(P.M.)	Program memory content

Figure 1-20. Data Bus Content During 4040-Only Instruction's Execution.

CHAPTER 2
PROGRAMMING THE MCS-40™ MICROCOMPUTER SYSTEM

ALSO REFER TO 4004 AND 4040 ASSEMBLY LANGUAGE PROGRAM MANUAL

Writing sequences of instructions for a computer is known as programming. To be able to program a computer effectively, the programmer must understand the action of each of the machine instructions. (The instruction set of the MCS-40 microcomputer is described in detail in Chapter 1.)

Each machine instruction manipulates data in some way. The data may be the contents of the program counter which indicates where the next instruction is to be found, the contents of one of the CPU registers, accumulator, or carry flip-flop, the contents of RAM or ROM, or the signals at a port.

Programming is probably most easily learned by use of examples. In the pages that follow, a number of sample program segments are described. In general, the examples are shown in order of increasing complexity. These examples have been chosen to illustrate such techniques as the use of the I/O ports, basic program loops, multiple precision arithmetic, and the use of subroutines. When reviewing these examples refer frequently to the instruction definitions.

Example #1

Consider the case where it is desired to test the status of a single switch connected to a 4004 CPU on the test input (pin 10). A jump on condition instruction (JCN) can be used to perform this test. The address jumped to must be in the same page (the same 256-byte segment of program memory). Suppose the JCN instruction: JCN TEST, 16 (2 word instruction) is stored at ROM memory locations 2 and 3. The instruction would look as follows:

	OPR	OPA
Location #2	0001	0001
	(JCN)	(Jump if test signal = Logic "0")
Location #3	0001	0000
	(Jump to ROM memory Location #16)	

When this instruction is executed, if the switch connects a logic "0" (V_{SS}) to the test pin of the CPU, the program counter in the address register in the CPU will jump to location 16. (That is, the next instruction to be executed would be fetched from ROM Memory location 16.) If the switch had been connected to a logic "1" (V_{DD}), the program counter would not jump but would be incremented by 1 and hence the instruction in ROM memory location 4 would be executed next. Thus, the switch status can be tested simply with one instruction. Furthermore, if it were desired to jump if a test signal equalled a logic "1", the JCN instruction could be coded:

	OPR	OPA	
Location #2	0001	1001	Inverted jump condition
Location #3	0001	0000	

In this case the invert condition bit C_1 is used to indicate a jump is to be made on a logic "1" on the test signal.

If more switches are required, a ROM port may be used as shown in the next example.

Example #2

Consider the case where it is desired to test the status of a switch connected to the I/O port of ROM #2. To make access to the port, it is necessary to execute an SRC instruction (reference section on operation of the command lines and the SRC command in Chapter 1). The SRC instruction utilizes the contents of a pair of registers, which must contain the proper numbers to select the desired port. Register pairs may be most easily loaded using the FIM instruction.

Thus the sequence:

Mnemonic	Description
FIM 0, 20H	;Fetch immediate (direct) from ROM data (0010, 000) to index register pair 0. (20H refers to 20 Hexadecimal.)
SRC 0	;Send the contents of index register pair 0 to select a ROM port. The first 4 bits of data sent out at X_2 time (0010) select ROM #2 (4001).

```
RDR        ;Read the contents of the previously
             selected ROM (ROM #2) input port
             into the accumulator.
```

These instructions have the effect of loading the accumulator with the values appearing at ROM port #2. Individual bits may be tested by shifting them into the carry flip-flop and using a jump on condition instruction. In this manner up to 4 switches can be interrogated from one set of ROM input ports (4 of them).

> *NOTE: All comment statements are punctuated with ";" at the initial statement description entry.*

Example #3

Suppose a series of 10 clock pulses must be generated, perhaps to drive the clock line of a 4003 port expander. Let us assume that the output port of RAM #3 is to be used to output the clock signal. The high order 2 bits of data sent out at X_2 time during an SRC instruction selects the RAM chip. Hence 1100 (binary equivalent of 12) is required at X_2 to select RAM #3.

Since we must select the port on RAM #3 we will require

```
FIM  0, 0C0H

SRC  0
```

This pair of instructions sets up the desired port for use. To generate the clock pulses, we must alternately write a 1 and a 0 into the appropriate port bit. Let us assume that we will only use the high order bit of the port on RAM #3 and that it is initially set at zero (so that the program does not have to reset it). Furthermore, let us assume that we do not care about the other three bits of the port.

First let us set the accumulator to 0

```
LDM   0      ;Set accumulator to 0
```

We may then complement the high order bit of the accumulator by the sequence:

```
RAL        ;Rotate left (accumulator and carry)
CMC        ;Complement carry
RAR        ;Rotate right (accumulator and carry)
```

which achieves the operation by shifting the bit into the carry flip-flop, complementing it, and shifting it back.

An alternate way to complement the high order bit is to add 8 (binary 1000) to the accumulator. We may set the contents of one register, say register 15, to 8 by the sequence:

```
LDM   8      ;Load data DDDD (1000) to the
               accumulator.
XCH   15     ;Exchange contents of index register 15
               and accumulator
LDM   0      ;Load (0000) to accumulator
```

The first instruction loads the binary number 1000 into the accumulator and the second places the contents of the accumulator into register 15. Since the prior contents of register 15 are also placed in the accumulator, an LDM instruction is then executed to clear the accumulator.

Now the operation ADD 15 will add the binary value 1000 to the accumulator, because Register 15 contains the value 8.

Note the difference in how the LDM and the XCH and ADD instructions utilize the second half of the instruction. The LDM loads the accumulator with the value carried by the instruction, i.e., in binary code LDM 8 appears as 1101 1000 and loads the accumulator with 1000. However, the ADD and XCH select a register, and the contents of the register are used as data. That is, ADD 8 would add the contents of register 8 to the accumulator, not the value 8.

To generate the sequence of 10 clock pulses, one could repeat the following 4 instructions 10 times.

```
ADD   15     ;Add contents of register 15
               (1000 previously stored in
               the register) to accumulator
WMP          ;Write the contents of the
               accumulator into the
               previously selected RAM
               output port
ADD   15
WMP
```

(one clock pulse generated)

However, this would take some 40 instructions. The indexing operation available with the ISZ instruction allows a program loop to be repeated 10 times.

The ISZ instruction increments a selected register. If the register initially contained any value other than the value 15 (binary 1111) the instruction performs a JUMP to an address specified by the instruction. This address must be on the same page (within the same 256-byte segment of program memory) as the instruction immediately following the ISZ.

If, however, the register originally contained 15, the CPU will proceed to execute the next instruction in sequence.

By loading a register, say register 14, with the value 6, on the 10th execution of an ISZ, the processor will proceed to the next instruction in sequence rather than jump.

Execution of the ISZ does not affect the accumulator, so that the accumulator does not have to be "saved" prior to its execution.

The program sequence which performs the desired action is then:

Instruction #	Address Name	Mnemonic	OPA	Description
(1)		LDM	8	;Load 1000 to accumulator
(2)		XCH	15	;Exchange contents of index register 15 and accumulator
(3)		LDM	6	;Load 0110 to accumulator
(4)		XCH	14	;Exchange contents of index register 14 and accumulator
(5)		FIM	0	;Fetch immediate from ROM Data (1100 0000) to index
		12,	0	;register pair location 0
(6)		SRC	0	;Send address (contents of index register pair 0) to RAM
(7)		LDM	0	;Set accumulator to 0
(8)	LOOP	ADD	15	;Add contents of register 15 to accumulator
(9)		WMP		;Write contents of accumulator into RAM output port
(10)		ADD	15	;Add contents of Register 15 to accumulator
(11)		WMP		;Write contents of accumulator into RAM output port
(12)		ISZ	14,LOOP	;Increment contents of register 14. Go to ROM address ;A_2, A_1 (called Loop) if result $\neq 0$, otherwise skip.

Explanation of Program

(a) Instruction #1 and #2 – Loads the number 8 (1000) into index register number 15 (1111)

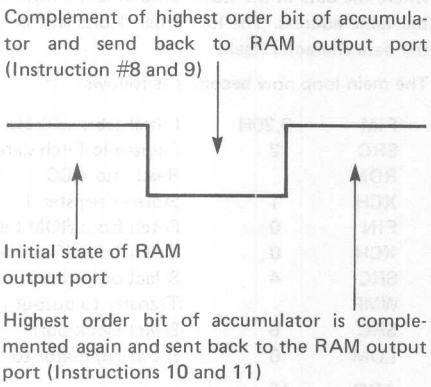
(b) Instruction #3 and #4 – Loads the number 6 (0110) into index register number 14 (1110)

(c) Instruction #5 – Fetches the address of the desired RAM and stores it in an index register pair

(d) Instruction #6 – Sends the stored address to the RAM bank and selects the desired RAM output port

(e) Instruction #7 – Initializes the accumulator to 0000.

(f) Instruction #8, 9, 10, and 11 – Generates one clock pulse as follows:



(g) Instruction #12 – The contents of Register 14 are incremented by 1 (0001). The number 7 (0111) is now stored in register 14. Since this result is not equal to zero, program control jumps to the address specified in the 2nd word of this instruction. In this case the address stored in the 2nd word is the address of instruction #8. The program then executes the next 4 instructions in sequence and generates a 2nd clock pulse. This sequence is repeated a total of 10 times, thus generating 10 clock pulses. On the 10th time when the contents of register 14 are incremented it goes to the value 0000 and the program skips to the next instruction in sequence and gets out of the loop.

Example #4

Clock pulse streams of the type derived above are often used to drive groups of 4003 shift registers (for a more detailed description of the 4003 shift register, see Chapter 5). It may often be desirable to transfer the contents of a RAM register to a group of 4 shift registers via two output ports as shown in Figure 2-1 (where the output ports are RAM output ports).

To operate this system, it is necessary to fetch a character from RAM and present it at an I/O port #2, then issue the clock pulse at an I/O port #1. This sequence requires three SRC commands, one for the RAM selection, one for I/O port selection, and one for I/O port #2 selection.

In addition, the location in RAM must be incremented each time to provide selection of the next character.

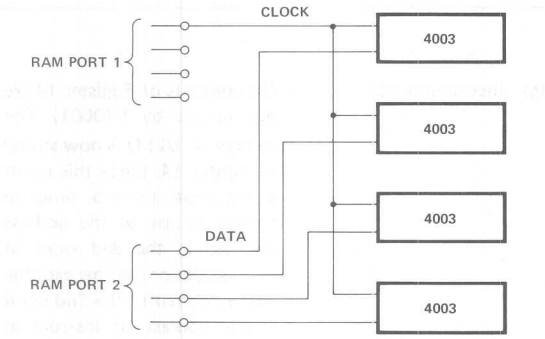


Figure 2-1. RAM Output Ports Driving Groups of Shift Registers

The main loop is then as follows:

```
LOOP:
SRC          ;Send address to select RAM
RDM          ;Read selected RAM character into
             ;accumulator
SRC          ;Send address to Port #2
WMP          ;Write contents of accumulator (previously
             ;read RAM character) into Port #2
SRC          ;Send address to Port #1
LDM   0      ;Set accumulator to "0"
ADD   15
WMP          ;Generate 1 clock pulse
ADD   15
WMP
INC          ;Increment by 1 the contents of the register
             ;pair holding the RAM address
ISZ   14,LOOP;Increment contents of register 1110.
             ;Jump if result ≠ 0, otherwise skip.
```

The loop above uses 3 pairs of registers for RAM and port selection, and two registers for temporary storage and indexing. The initialization must provide for loading each of these registers.

Example #5

The example above might be extended if, for example, the 4003's were driving seven segment LED displays. A 4 line to 7 segment code converter could be used for each display device driven. However, the ROM table lookup capability of an MCS-40 CPU can be utilized to advantage to save these converters. Suppose the LED displays are wired as shown in Figure 2-2, with each LED using two adjacent locations in each of the 4003's.

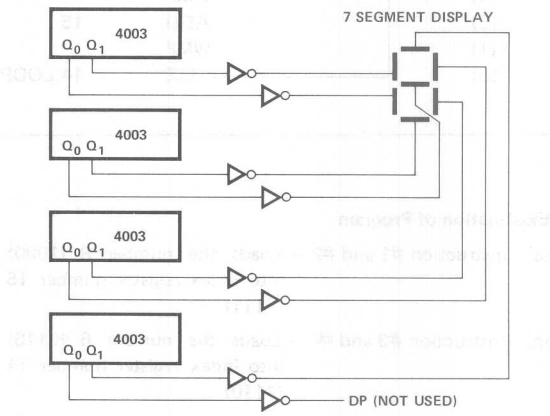


Figure 2-2. Shift Registers Driving Seven Segment LED Displays

The instruction FIN allows a ROM table to be accessed based on the contents of registers 0 and 1. To save register space, the fetched data may be loaded over the table addresses. The table address may be initialized by an FIM or by the sequence:

```
LDM
XCH
```

where the data in the LDM represents the high-order 4 bits of the table address. The low order 4 bits will be derived from the data character itself.

The main loop now becomes as follows:

```
FIM   0,20H  ;Initial table address
SRC   2      ;Prepare to fetch data character
RDM          ;Read into ACC
XCH   1      ;Store at register 1
FIN   0      ;Fetch from ROM table
XCH   0      ;Fetch 1st half of 7 segments
SRC   4      ;Select output port
WMP          ;Transfer to output port
SRC   6      ;Select clock port
LDM   0      ;Set accumulator to "0"
ADD   15
WMP          ;Generate one clock pulse
ADD   15
WMP
```

```
SRC     4       ;Select output port
XCH     1       ;Transfer 2nd half of display
WMP             ;Transfer to output port
SRC     6       ;Select clock port
LDM     0       ;Set accumulator to "0"
ADD     15      ┐
WMP             ├ ;Generate one clock pulse
ADD     15      ├
WMP             ┘
INC     3       ;Set next RAM character
ISZ     14      ;Test for No. of characters
```

Note that two data characters (4 bits each) are transferred for each digit to be displayed.

This loop must be initialized by setting the registers to their initial conditions. The following sequence of 4 instructions is sufficient:

```
FIM     2,  *   ;Select RAM register for display
FIM     6,  *   ;Initialize clock port selector
FIM     4,  *   ;Initialize output port selector
FIM     14, *   ;Initialize no. of digits and set reg.
                ;#15 to 1000.
```

*Immediate values loaded are dependent on specific port numbers of the system configuration and number of characters to be displayed.

Example #6

Proceeding with the example outlined above, suppose that the user finds it necessary to display the contents of a number of different RAM registers, at different places in the program. The sequence of instructions could be used whenever this was necessary. However, by making the entire sequence a "subroutine", the user can call out the sequence each time it is needed with only a JMS instruction.

The JMS utilizes the address push down stack. When a JMS is executed, the program counter is pushed into the address register push-down stack and is reloaded with the address to which the jump is to take place. Execution will proceed from this new location. This push-down stack operates as follows:

1. Each time a JMS is executed, all addresses saved in the stack are pushed down 1 level. The last value of the program counter is loaded into the top of the stack. The program counter value corresponds to the instruction immediately following the JMS.
2. The BBL (and also the BBS in the 4040) instruction raise every entry in the stack one level, with the top value in the stack entering the program counter.

In the example shown, if the RAM register to be transferred to the display is different in different parts of the program, the FIM which selects the RAM register should not be made part of the subroutine. The subroutine would then include the three FIM instructions followed by the main loop and terminated by a BBL.

To display any RAM register from any point in the program, the programmer need use only 4 bytes of ROM:

```
FIM     Reg Pair, Data (Byte)
JMS
```

The FIM loads the RAM register address into the register pair to be used by an SRC instruction and the JMS calls the subroutine.

Example #7 Interpretive Mode

Interpretive mode programming may be used to reduce the amount of ROM required to implement a particular system function. In this mode, data words fetched from ROM or RAM are treated as instructions of a computer which might be quite different than the MCS-40™ microcomputer. The MCS-40 program "interprets" the data, using it to call appropriate subroutines which simulate the instructions of the different computer. In effect another computer architecture is simulated.

In the interpretive mode, the instructions of the simulated computer (pseudo instructions) may be derived from RAM or ROM. If the instructions are in RAM, they are fetched from RAM via the normal RAM operations (SRC, RDM), using a simulated program counter to maintain the address. The JIN instruction is often useful for interpreting the fetched instruction. (The address for the JIN is computed from the fetched pseudo instruction. Each address value is the location of a JMP, or JMS to an appropriate routine, or the routine itself.)

When fetching pseudo instructions from ROM, the FIN is used. As the FIN instruction must be located in the same 256 byte programming segment (page) as the fetched data, one cannot use all 256 8 bit bytes of a ROM for pseudo instructions. It is sufficient to allow an FIN followed by a BBL on the program memory page. Thus, up to 254 bytes of each page can be used for pseudo instructions. The simulated program counter must correspond to this address structure. If the FIN and BBL instructions are located in the first two locations of the program memory page, the 254 step program address counter can be implemented by initializing a 4001 ROM chip address to location 2 rather than location 0. If the interpretive mode program exceeds 254 instructions, the program control routine must determine the proper page to find the next pseudo instruction. The instruction is then fetched by a JMS to address 0 of the appropriate page. Refer to the 4004 and 4040 Assembly Language Programmer's Manual for further detials.

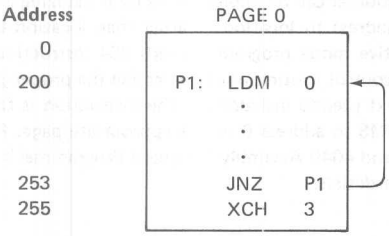
PROGRAMMING TECHNIQUES

This section describes some techniques which may be of help to the MCS-40™ programmer.

CROSSING PAGE BOUNDARIES

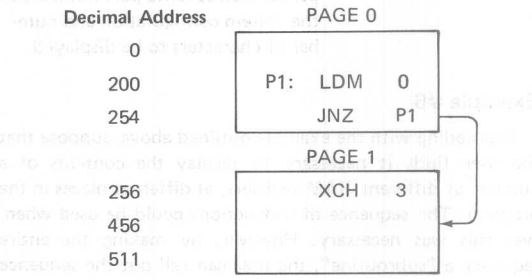
As described in Chapter 1, programs are held in either ROM, PROM, or program RAM, all of which are divided into pages. Each page consists of 256 8-bit locations. Addresses 0 through 255 comprise the first page, 256-511 comprise the second page, and so on.

In general, it is good programming practice to *never allow program flow to cross a page boundary except by using a JUN or JMS instruction.* The following example will show why this is true. Suppose a program in memory appears as below:



If the accumulator is non-zero when the JNZ is executed, program control will be transferred to location 200, as the programmer intended.

Suppose now that an error discovered in the program requires that a new instruction be inserted somewhere between locations 200 and 253. The program would now appear as follows:



Since the JNZ is now located in the last two locations of a page, it functions differently. Now if the accumulator is non-zero when the JNZ is executed, program control will be erroneously transferred to location 456, causing invalid results.

Since both the JUN and JMS instructions use 12 bit addresses to directly address locations on any page of memory, only these instructions should be used to cross page boundaries.

SUBROUTINES

Frequently, a group of instructions must be repeated many times in a program. The group may be written "n" times if it is needed at "n" different points in a program, but better economy can be obtained by using subroutines.

A subroutine is coded like any other group of assembly language statements, and is referred to by its name, which is the label of the first instruction. The programmer references a subroutine by writing its name in the operand field of a JMS instruction. When the JMS is executed, the address

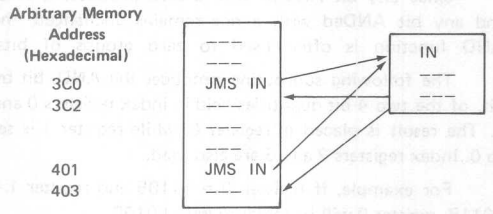
of the next sequential instruction after the JMS is written to the address stack, and program execution proceeds with the first instruction of the subroutine. When the subroutine has completed its work, a BBL instruction is executed, which loads a value into the accumulator and causes an address to be read from the stack into the program counter, causing program execution to continue with the instruction following the JMS. Thus, one copy of a subroutine may be called from many different points in memory, preventing duplication of code. Note also that since the address stack and the JMS instruction use 12-bit addresses, calling programs and subroutines may be located anywhere in program memory (they need not be on the same page in memory).

Example:

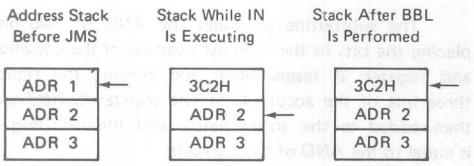
Subroutine IN increments an 8 bit number passed in index register 0 and 1 and then returns to the instruction following the last JMS instruction executed.

```
Label   Code    Operand
IN:     XCH     1       ; Reg 1 to Accum.
        IAC             ; Increment value and
                        ; produce carry
        XCH     1       ; Restore reg 1.
        JNC     NC      ; Jump if Carry = 0.
        INC     0       ; Increment high order
                        ; 4 bits
NC:     BBL     0       ; Return
```

Assume IN appears as follows:



When the first JMS is executed, address 3C2H is written to the address stack, and control is transferred to IN. Execution of the BBL statement will cause the address 3C2H to be read from the stack and placed in the program counter, causing execution to continue at 3C2H (since the JMS occupies two bytes).



When the second JMS is executed, address 403H is written to the stack, and control is again transferred to IN. This time, the BBL will cause execution to resume at 403H.

Note that IN could have called another subroutine during its execution, causing another address to be written to the stack. For a 4004, this can occur only up to three levels, however, since the stack can hold only three addresses. Beyond this point, some addresses will be overwritten and BBL's will transfer program control to incorrect addresses.

4040 ONLY:

Since the stack in the 4040 contains 7 registers, subroutines may be nested up to seven levels deep.

BRANCH TABLE PSEUDO-SUBROUTINE

Suppose a program consists of several separate routines, any of which may be executed depending upon some initial condition (such as a bit set in the accumulator). One way to code this would be to check each condition sequentially and branch to the routines accordingly as follows:

```
CONDITION = CONDITION 1 ?
IF YES BRANCH TO ROUTINE 1
CONDITION = CONDITION 2 ?
IF YES BRANCH TO ROUTINE 2
        .
        .
        .
BRANCH TO ROUTINE N
```

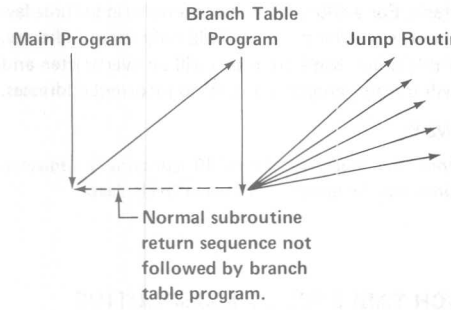
A sequence as above is inefficient, and can be improved by using a branch table.

The logic at the beginning of the branch table program computes an index into the branch table. The branch table itself consists of a list of starting addresses for the routines to be selected. Using the table index, the branch table program loads the selected routine's starting address into a register pair and executes a "jump indirect" to that address. For example, consider a program that executes one of five routines depending upon which bit (possibly none) of the accumulator is set:

```
Jump to routine 0 if accumulator = 0000 B
Jump to routine 1 if accumulator = 0001 B
Jump to routine 2 if accumulator = 0010 B
Jump to routine 3 if accumulator = 0100 B
Jump to routine 4 if accumulator = 1000 B
```

A program that provides the above logic is given at the end of this section. The program is termed a "pseudo-subroutine" because it is treated as a subroutine by the programmer, (i.e., it appears just once in memory), but it is entered via a regular "jump" instruction rather than

via a JMS instruction. This is possible because the branch routines control subsequent execution, and will never return to the instruction following JMS.




```
Label  Code   Operand
ST:    KBP                ; Convert Accum. to
                          ; branch table index.
       IAC                ; If = 1111B, ERROR
       JZ     ERR         ; Jump if IAC produced
                          ;.zero.
       DAC                ; O.K., restore Accum.
       FIM    0,BTL       ; Regs 0 and 1 =
                          ; address of branch
                          ; table.
       CLC                ; Carry = 0
       ADD    1           ; Add index to branch
                          ; table address
       XCH    1           ; Store back to reg 1
       JNC    NC          ; Jump if no carry
       INC    0           ; If carry, increment
                          ; reg. 0.
NC:    FIN    0           ; Regs 0 and 1 - add-
                          ; ress of routine.
       JIN    0           ; Jump to correct
                          ; routine.
                .
                .
                .
BTL:   DB     RT0 AND 0FFH ; Branch table. Each
       DB     RT1 AND 0FFH ; entry is an 8-bit
       DB     RT2 AND 0FFH ; address.
       DB     RT3 AND 0FFH
       DB     RT4 AND 0FFH
                .
                .
                .
ERR:   ---                ; Error handling
                          ; routine.
```

NOTE: Since FIM, FIN, and JIN operate with 8-bit addresses, *routines ST, BTL, and RT0 through RT4 must all reside in the same page of memory.*

If the accumulator held 0100B when location ST was reached, the KBP would convert it to 0011B. The 8 bit address at BTL + 3 would therefore be loaded into registers 0 and 1, and the JIN would cause program control to be transferred to routine RT3.

LOGICAL OPERATIONS

This section gives three subroutines which produce the logical operations "AND", "OR", and "XOR" (exclusive-OR). Note that for 4040 programming there are special hardware instructions for "AND" and "OR" operations on certain index registers (see instruction set summary in Chapter 1).

Logical "AND"

The AND function of two bits is given by the following truth table:

	0	1
0	0	0
1	0	1

Since any bit ANDed with a zero produces a zero, and any bit ANDed with a one remains unchanged, the AND function is often used to zero groups of bits.

The following subroutine produces the AND, bit by bit, of the two 4-bit quantities held in index registers 0 and 1. The result is placed in register 0, while register 1 is set to 0. Index registers 2 and 3 are also used.

For example, if register 0 = 1110B and register 1 = 0011B, register 0 will be replaced with 0010B.

```
        1110B
AND     0011B
        ----
        0010B
```

The subroutine produces the AND of two bits by placing the bits in the leftmost position of the accumulator and register 2, respectively, and zeroing the rightmost three bits of the accumulator and register 2. Register 2 is then added to the accumulator, and the resulting carry is equal to the AND of the two bits.

```
Label   Code    Operand
ANDS:   FIM     2,11    ; Reg 2 = 0, reg 3 = 11
L1:     LDM     0       ; Set ACC = 0
        XCH     0       ; Reg 0 data to ACC;
                        ; reg 0 = 0
        RAL             ; 1st 'AND' bit to carry
        XCH     0       ; Save shifted data in
                        ; reg 0; ACC = 0
        ISZ     3,L2    ; Done if reg 3 = 0
        JUN     L3
L2:     RAR             ; Bit of reg 0 is alone in
                        ; ACC
        XCH     2       ; Save 1st 'AND' bit in
                        ; reg 2
        XCH     1       ; Get bit of reg 1
        RAL             ; Left bit to carry
        XCH     1       ; Save shifted data in
                        ; reg 1
        RAR             ; 2nd 'AND' bit to ACC
        ADD     2       ; 'ADD' gives 'AND' of
                        ; the 2 bits in carry
        JUN     L1
L3:     BBL     0       ; Return to main pro-
                        ; gram.
```

Logical "OR"

The OR function of two bits is given by the following truth table:

	0	1
0	0	1
1	1	1

Since any bit ORed with a one produces a one, and any bit ORed with a zero remains unchanged, the OR function is often used to set groups of bits to one.

The following subroutine produces the OR, bit by bit, of the two 4-bit quantities held in index registers 0 and 1. The result is placed in register 0, while register 1 is set to 0. Index registers 2 and 3 are also used.

For example, if register 0 = 0100B and register 1 = 0011B, register 0 will be replaced with 0111B.

```
      0100B
OR    0011B
      ----
      0111B
```

The subroutine produces the OR of two bits by placing the bits in the leftmost position of the accumulator and register 2, respectively, and zeroing the rightmost three bits of the accumulator and register 2. Register 2 is then added to the accumulator. If the resulting carry = 1, the OR of the two bits = 1. If the resulting carry = 0, the OR of the two bits is equal to the leftmost bit of the accumulator.

```
Label   Code    Operand
ORS:    FIM     2,11    ; Reg 2 = 0, reg 3 = 11
L1:     LDM     0       ; Set ACC = 0
        XCH     0       ; Reg 0 data to ACC;
                        ; reg 0 = 0
        RAL             ; 1st 'OR' bit to carry
        XCH     0       ; Save shifted data in
                        ; reg 0; ACC = 0
        ISZ     3,L2    ; Done if reg 3 = 0
        JUN     L3
L2:     RAR             ; Bit of reg 0 is alone in
                        ; ACC
        XCH     2       ; Save 1st 'OR' bit in
                        ; reg 2
        LDM     0       ; Get bit in reg 1; set
                        ; ACC = 0
        XCH     1
        RAL             ; Left bit to carry
        XCH     1       ; Save shifted data in
                        ; reg 1
        RAR             ; 2nd 'OR' bit to ACC
        ADD     2       ; Produce the OR of the
                        ; bits
        JC      L1      ; Jump if carry = 1
                        ; Otherwise 'OR' = left
        RAL             ; bit of accumulator
        JUN     L1      ; Transmit to carry by
L3:     BBL     0       ; RAL
                        ; Return to main program.
```

Logical "XOR" Exclusive-OR

The XOR (exclusive-OR) function of two bits is given by the following truth table:

	0	1
0	0	1
1	1	0

Since the exclusive OR of two equal bits produces a zero and the exclusive OR of two unequal bits produces a one, the exclusive OR function can be used to test two quantities for equality. If the quantities differ in any bit position, a one will be produced in the result.

The following subroutine produces the exclusive-OR of the two 4-bit quantities held in index registers 0 and 1. The result is placed in register 0, while register 1 is set to 0. Index registers 2 and 3 are also used.

For example if register 0 = 0011B and register 1 = 0010B, register 0 will be replaced with 0001B.

```
      0011B
XOR   0010B
      ----
      0001B
```

The subroutine produces the XOR of two bits by placing the bits in the leftmost position of the accumulator and register 2, respectively, and zeroing the rightmost three bits of the accumulator and register 2. Register 2 is then added to the accumulator. The XOR of the two bits is then equal to the leftmost bit of the accumulator.

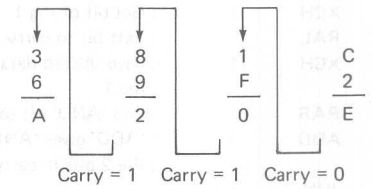
```
Label   Code    Operand
XORS:   FIM     2,11    ; Reg 2 = 0, reg 3 = 11
L1:     LDM     0       ; Set ACC = 0
        XCH     0       ; Reg 0 data to ACC;
                        ; reg 0 = 0
        RAL             ; 1st XOR bit to carry
        XCH     0       ; Save shifted data in
                        ; reg 0; ACC = 0
        ISZ     3,L2    ; Done if reg 3 = 0
        JUN     L3
L2:     RAR             ; Bit of reg 0 is alone
                        ; in ACC
        XCH     2       ; Save 1st XOR bit in
                        ; reg 2
        LDM     0       ; Get bit in reg 1; set
                        ; ACC = 0
        XCH     1
        RAL             ; Left bit to carry
        XCH     1       ; Save shifted data in
                        ; reg 1
        RAR             ; 2nd 'XOR' bit to ACC
        ADD     2       ; Produce the XOR of
                        ; the bits
        RAL             ; XOR = left bit of
                        ; Accum; transmit
        JUN     L1      ; to carry by RAL.
L3:     BBL     0
```

MULTI-DIGIT ADDITION

The carry bit may be used to add unsigned data quantities of arbitrary length. Consider the following addition of two 4-digit hexadecimal numbers:

```
  381C
+ 69F2
------
  A20E
```

This addition may be performed by setting the carry bit = 0, adding the two low-order digits of the numbers, then adding the resulting carry to the two next higher order digits, and so on:



The following subroutine will perform a sixteen digit addition, making these assumptions:

The two numbers to be added are stored in DATA RAM chip 0, registers 0 and 1.

The numbers are stored with the least significant digit first (in character 0).

The result will be stored least significant digit first in register 1, replacing the contents of register 1.

Index register 8 will count the number of digits (up to 16) which have been added.

DATA RAM CHIP 0 BEFORE ADDITION

	Status Characters																			
Register 0	C	1	8	3	0	0	0	0	0	0	0	0	0	0	0	0				
1	2	F	9	6	0	0	0	0	0	0	0	0	0	0	0	0				
2																				
3																				

DATA RAM CHIP 0 AFTER ADDITION

Register 0	C	1	8	3	0	0	0	0	0	0	0	0	0	0	0	0				
1	E	0	2	A	0	0	0	0	0	0	0	0	0	0	0	0				
2																				
3																				

```
Label   Code    Operand
AD:     FIM     4,0     ; Reg pair 4 = RAM chip
                        ; 0 of reg 0
        FIM     6,10H   ; Reg pair 6 = RAM chip
                        ; 0 of reg 1
        CLB             ; Set carry = 0
        XCH     8       ; Set digit counter = 0
AD1:    SRC     4       ; Select RAM reg 0
        RDM             ; Read digit to
                        ; accumulator
        SRC     6       ; Select RAM reg 1
        ADM             ; Add digit + carry to
                        ; accumulator
        WRM             ; Write result to reg 1
        INC     5       ; Address next char.
                        ; of RAM reg 0
        INC     7       ; Address next char.
                        ; of RAM reg 1
        ISZ     8,AD1   ; Branch if digit
                        ; counter <16
                        ; (non-zero)
OVR:    BBL     0
```

When location OVR is reached, RAM register 1 will contain the sum of the two 16 digit numbers arranged from low order digit to high order digit. (The reason multi-digit numbers are arranged this way is that it is easier to add numbers from low order to high order digit, and it is easier to increment addresses than to decrement them).

The first time through the program loop, index register pair 4 (index register 4 and 5) contains 0 and index register pair 6 (index registers 6 and 7) contains 16, referencing the first data characters of DATA RAM registers 0 and 1, respectively.

On succeeding repetitions of the loop, index registers 5 and 7 are incremented, referencing sequential data characters, until all 16 digits have been added.

MULTI-DIGIT SUBTRACTION

The carry bit may be used to subtract unsigned data quantities of arbitrary length. Consider the following subtraction of two 4-digit hexadecimal numbers:

```
  54BA
- 14F6
------
  3FC4
```

This subtraction may be performed by first setting the carry bit = 1. Then, for each pair of digits, the program must complement the carry bit and perform the subtraction. By this process, the carry bit will adjust the differences, taking into account any borrows which may have occurred.

This process applied to the above subtraction proceeds as follows:

1) Set carry bit = 1.

2) Complement carry bit. Carry now = 0.

3) Subtract low order digits:

```
  A = 1010B
  6 = 1001B
carry =    1
  1|0100B = 4
```

4) Complement resulting carry. Carry now = 0.

5) Subtract next digits:

```
  B = 1011B
  F = 0000B
carry =    1
  0|1100B = CH
```

6) Complement resulting carry. Carry now = 1.

7) Subtract next digits:

```
  4 = 0100B
  4 = 1011B
carry =    0
  0|1111B = FH
```

8) Complement resulting carry. Carry now = 1.

9) Subtract next digits:

```
  5 = 0101B
  1 = 1110B
carry =    0
  1|0011B = 3
```

Thus the correct result, 3FC4H, is obtained. The following subroutine will perform a sixteen digit subtraction, making these assumptions:

As in the example on page 2-7, the two numbers are stored in DATA RAM chip 0, registers 0 and 1 (register 1 containing the subtrahend). The numbers are stored with the least significant digit in character 0, and the result is stored back into register 1. Index register 8 will count the number of digits (up to 16) which have been subtracted.

```
Label   Code    Operand
SB:     FIM     4,0       ; Reg pair 4 = RAM
                          ; chip 0 reg 0
        FIM     6,10H     ; Reg pair 6 = RAM
                          ; chip 0 reg 1
        CLB
        XCH     8         ; Set digit counter = 0
        STC               ; Set carry = 1
SB1:    CMC               ; Complement carry bit
        SRC     4         ; Select RAM reg 0
        RDM               ; Read digit to
                          ; accumulator
        SRC     6         ; Select RAM reg 1
        SBM               ; Subtract digit and
                          ; carry from
                          ; accumulator
        WRM               ; Write result to reg 1
        INC     5         ; Address next char.
                          ; of RAM reg 0
        INC     7         ; Address next char.
                          ; of RAM reg 1
        ISZ     8,SB1     ; Branch if digit
                          ; counter < 16
                          ; (non-zero).
OV:     BBL     0
```

When location OV is reached, RAM register 1 will contain the difference of the two 16 digit numbers. Note that the carry bit from the previous subtraction is complemented by the CMC instruction each time through the program loop.

DECIMAL ADDITION

Each 4 bit data quantity may be treated as a decimal number as long as it represents one of the decimal digits from 0 through 9, and does not contain any of the bit patterns representing the hexadecimal digits A through F. In order to preserve this decimal interpretation when performing addition, the value 6 must be added to the accumulator whenever an addition produces a result between 10 and 15. This is because each 4 bit data quantity can hold 6 more combinations of bits than there are decimal digits.

The DAA (decimal adjust accumulator) instruction is provided for this purpose. Also, to permit addition of multi-digit decimal numbers, the DAA adds 6 to the accumulator whenever the carry bit is set indicating a decimal carry from previous additions. The carry bit is unaffected unless the addition of 6 produces a carry, in which case the carry bit is set.

To perform the decimal addition:

```
  468
+ 329
-----
  798
```

the process works as follows.

1) Clear the carry and add the lowest-order digits

```
    9 = 1001B
    9 = 1001B
carry =    0
      1|0010B
       ↑
       └── Carry = 1
```

2) Perform a DAA operation, which will add 6 to the accumulator. Since no carry is produced by this operation, the carry bit is left unaffected, remaining = 1.

```
Accumulator = 0010B
          6 = 0110B
      carry =    0
        0|1000B = 8
```

3) Add the next two digits.

```
    6 = 0110B
    2 = 0010B
carry =    1
      0|1001B = 0
       ↑
       └── Carry = 0
```

4) Perform a DAA operation. Since the accumulator is not greater than 9 and the carry is not set, no action occurs.

5) Add the next two digits:

```
    4 = 0100B
    3 = 0011B
carry =    0
      0|0111B = 7
       ↑
       └── Carry = 0
```

6) Perform a DAA operation. Again, no action occurs. Thus the correct decimal result 798 is generated in three 4 bit data characters.

A subroutine which adds two 16 digit decimal numbers, then, is exactly analagous to the 16 digit hexadecimal addition subroutine on page 2-12, and may be produced by inserting the instruction DAA after the ADM instruction of that example.

DECIMAL SUBTRACTION

Each 4 bit data quantity may be treated as a decimal number as long as it represents one of the decimal digits 0 through 9. The TCS (transfer carry subtract) and DAA (decimal adjust accumulator) may be used to subtract two decimal numbers and produce a decimal number. In fact, the TCS instruction permits subtraction of multi-digit decimal numbers.

The process consists of generating the ten's complement of the subtrahend digit (the difference between the subtrahend digit and 10 decimal), and adding the result to the minuend digit. For instance, to subtract 2 from 7, the ten's complement of 2 (10 − 2 = 8) is added to 7, producing 15 decimal which, when truncated to a 4 bit quantity gives 5 (the required result). If a borrow was generated by the previous subtraction, the 9's complement of the subtrahend digit is produced to compensate for the borrow.

In detail, the procedure for subtracting one multi-digit decimal number from another is as follows:

1) Set the carry bit = 1 indicating no borrow.
2) Use the TCS instruction to set the accumulator to either 9 or 10 decimal.
3) Subtract the subtrahend digit from the accumulator, producing either the 9's or 10's complement.
4) Set the carry bit = 0.
5) Add the minuend digit to the accumulator.
6) Use the DAA instruction to make sure the result in the accumulator is in decimal format, and to indicate a borrow in the carry bit if one occurred.

   Save this result.
7) If there are more digits to subtract, go to step 2.

   Otherwise stop.

Example:

Perform the decimal subtraction:

$$\begin{array}{r} 51 \\ -\,38 \\ \hline 13 \end{array}$$

1) Set carry = 1.
2) TCS sets accumulator = 1010B and carry = 0.
3) Subtract the subtrahend digit 8 from the accumulator.

$$\begin{array}{rl} \text{Accumulator} = & 1010\text{B} \\ \overline{8} = & 0111\text{B} \\ \overline{\text{carry}} = & \quad 1 \\ & 0010\text{B} \end{array}$$

4) Set carry = 0.
5) Add minuend digit 1 to accumulator.

$$\begin{array}{rl} \text{Accumulator} = & 0010\text{B} \\ 1 = & 0001\text{B} \\ \text{carry} = & \quad 0 \\ 0 & 0011\text{B} = 3 \end{array}$$

Carry = 0

6) DAA leaves accumulator = 3 = first digit of result, and carry = 0, indicating that a borrow occurred.
7) TCS sets accumulator = 1001B and carry = 0.
8) Subtract the subtrahend digit 3 from the accumulator.

$$\begin{array}{rl} \text{Accumulator} = & 1001\text{B} \\ \overline{3} = & 1100\text{B} \\ \overline{\text{carry}} = & \quad 1 \\ & 0110\text{B} \end{array}$$

9) Set carry = 0.
10) Add minuend digit 5 to accumulator.

$$\begin{array}{rl} \text{Accumulator} = & 0110\text{B} \\ 5 = & 0101\text{B} \\ \text{carry} = & \quad 0 \\ 0 & 1011\text{B} \end{array}$$

Carry = 0

11) DAA adds 6 to accumulator and sets carry = 1, indicating that no borrow occurred.

$$\begin{array}{rl} \text{Accumulator} = & 1011\text{B} \\ 6 = & 0110\text{B} \\ 1 & 0001\text{B} = 1 = \text{Second digit of result.} \end{array}$$

Carry = 1

Therefore the result of subtracting 38 from 51 is 13.

The following subroutine will subtract one 16 digit decimal number from another, using the following assumptions.

The minuend is stored least significant digit first in DATA RAM chip 0, register 0.

The subtrahend is stored least significant digit first in DATA RAM chip 0, register 1.

The result will be stored least significant digit first in DATA RAM chip 0, register 0, replacing the minuend.

Index register 8 will count the number of digits (up to 16) which have been subtracted.

```
Label   Code    Operand
SD:     FIM     4,0       ; Reg pair 4 = RAM
                          ; chip 0, reg 0
        FIM     6,10H     ; Reg pair 6 = RAM
                          ; chip 0, reg 1
        CLB
        XCH     8         ; Set digit counter = 0
        STC               ; Set carry = 1
SD1:    TCS               ; Accumulator = 9
                          ; or 10
        SRC     6         ; Select RAM reg 1
        SBM               ; Produce 9's or 10's
                          ; complement
        CLC               ; Set carry = 0
        SRC     4         ; Select RAM reg 0
        ADM               ; Add minuend to
                          ; accumulator
        DAA               ; Adjust accumulator
        WRM               ; Write result to reg 0
        INC     5         ; Address next char.
                          ; of RAM reg 0
        INC     7         ; Address next char.
                          ; of RAM reg 1
        ISZ     8,SD1     ; Branch if digit
                          ; counter < 16
                          ; (non-zero)
DN:     BBL     0
```

Floating Point Numbers

The 4002 RAM may be used for storing and fetching a floating point decimal number. The 4002 RAM has 4 registers, each with twenty 4 bit characters subdivided into 16 main memory characters and 4 status characters (320 bits total). Each register is capable of storing a 20 digit, unsigned, fixed point, binary-coded decimal (BCD) number. A more practical use for the register is the storage of a signed, floating point, BCD number having a 16 digit mantissa (fraction) and a 2 digit exponent.

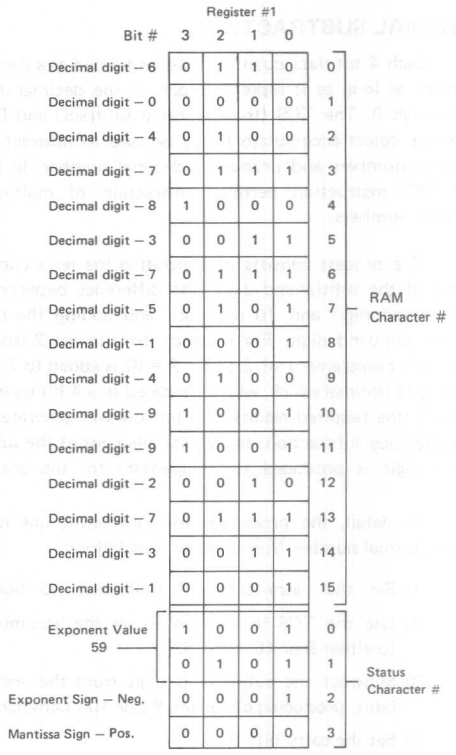
Consider the number

$$+.1372994157387406 \times 10^{-59}$$

Mantissa (16 digits) — Exponent (2 digits)

Storage is required for both the sign of the mantissa (in this case positive) and the sign of the exponent (in this case negative), 16 digits of mantissa and 2 digits of exponent. The 4 status characters of the register can be used to hold the signs (in this case a "1" represents minus – this definition is completely arbitrary and is completely up to the user) and the 2 digit exponent. The 16 main memory characters are used to hold the 16 digit mantissa.

For example, let's store the previously shown number in RAM bank #2, chip number #3, register #1. It would be stored in the 4002 as follows:



The following instructions would be used to fetch character #6, the signs, and exponent value:

	Mnemonic		Machine Language OPR	OPA
Select Bank #2	LDM	2	1101	0010
	DCL		1111	1101
Select Chip #3, Register #1 Character #6	FIM	4	0010	1000
	13,	6	11 (Chip #3) 01 (Register #1)	0110 (Main memory Character #6)
	SRC	4	0010	1001
Fetch the Mantissa sign From status Character #3 to Register #10 in the CPU	RD3		1110	1111
	XCH	10	1011	1010
Fetch the exponent sign From status Character #2 to Register #11 in the CPU	RD2		1110	1110
	XCH	11	1011	1011
Fetch the exponent from status Character #1 and #0 to Register #12 and #13 respectively	RD1		1110	1101
	XCH	12	1011	1100
	RDO		1110	1100
	XCH	13	1011	1101
Fetch the previously selected main memory Character #6 (which stored the decimal digit 7 to the accumulator	RDM		1110	1001

EXTENDED BANK SWITCHING

The following describes how additional ROM banks can be added to an MCS-40 system to expand ROM program memory beyond one ROM bank of 4K bytes for a 4004 or beyond two ROM banks of 4K bytes each for a 4040.

When the ROM bank is to be switched, the RAM output port has one line either set or reset. The drawing assumes SET (logical "1", MCS-40™ definition) to select BANK 1 and RESET (logical "0") to select BANK 0. The normal condition shall be BANK 0 selected.

The RAM port output steers the CM-ROM line to one of two separated output lines labeled CM-ROM BANK 0 and CM-ROM BANK 1. While the logic shown is a TTL implementation, an equivalent circuit at MOS levels may be built.

The circuit will functionally switch banks under software control by utilizing the WMP command for writing accumulator contents to the RAM port (a more detailed description of the programming sequence will be presented later). The next instruction after the WMP will execute from the bank selected. The bank selection remains in effect until another WMP for that port is issued (port value changed). Any subsequent instructions execute from the newly selected bank. The operation is similar to the DCL command for RAM BANK selection.

Because the program counter contents are incremented, execution of a command after the bank switch begins at PC + 1 but from the new ROM bank, not the ROM bank which issued the switch command.

I.E.

BANK	PC	PROGRAM
0	00	.
0	01	.
0	02	WMP ← SWITCH HERE
1	03	.
1	04	.
1	05	.

BANK	PC	PROGRAM
1	06	.
1	07	WMP ← SWITCH BACK
0	08	.
0	09	.
0	0A	.
0	0B	.

To avoid unnecessary gaps in either bank, an established switch point should be made. Since it is assumed that more than 16 pages are necessary, then BANK 0 page 15 location FF is always available. Also, location FF of the last used page of BANK 1 is always available. Therefore, two "natural" switch points for bank switching exist. These "natural" points are easily seen if we assume 16 pages in each bank. Then a programming sequence might look like . . .

	BANK	PC	PROGRAM
	0	FFB	.
	0	FFC	.
	0	FFD	.
	0	FFE	.
Last byte, last page BANK 0	0	FFF	WMP ← SWITCH HERE
	1	000	.
	1	001	.
	1	002	.
	1	003	.
	1		.
	1	etc.	.
	1		.
	1	FFC	.
	1	FFD	.
	1	FFE	.
Last byte, last page BANK 1	1	FFF	WMP ← SWITCH BACK
	0	000	.
	0	001	.

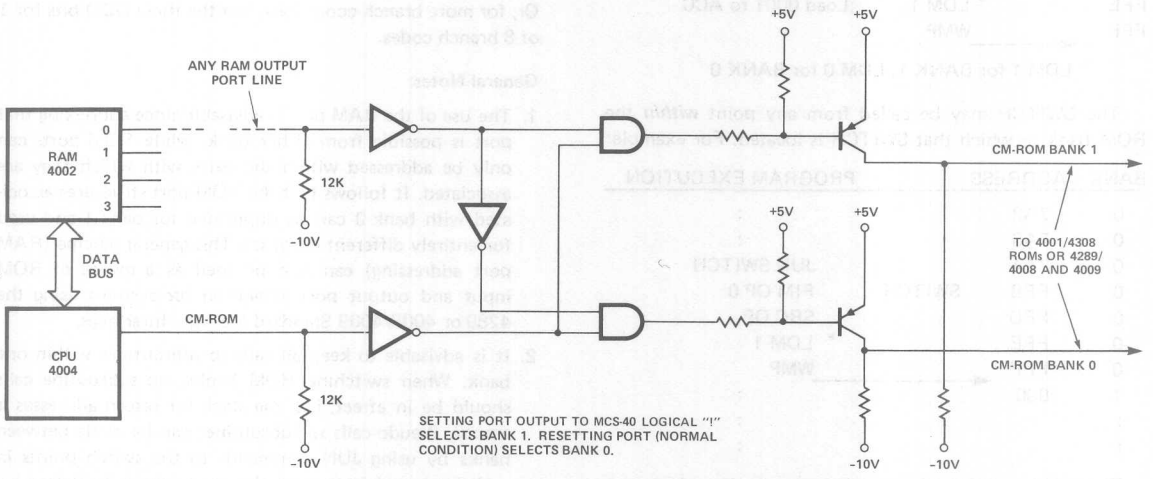


Figure 2-3. ROM Bank Selection Logic.

This is an "end around" switch and the beginning locations in ROM bank 0 must be reserved to "recognize" the switch back to bank 0 from bank 1. The recognition may be as simple as a jump back to the program beginning in BANK 0. For example:

BANK	PC	PROGRAM	
1	:	:	
1	FFF	WMP	SWITCH HERE
0	000	JUN	START
0	010		(address of START)

It may also cause a jump to any address within BANK 0.

If less than 16 pages are used in bank 1, then as an example:

	BANK	PC	PROGRAM	
	0	FFD	:	
	0	FFE	:	
	0	FFF	WMP	SWITCH HERE
	1	000	.	
	1	001	.	
	1	002	.	
	1	6FC	:	
	1	6FD	:	
	1	6FE	:	
Last byte, last page (only 0-6 used in Bank 1)	1	6FF	WMP	SWITCH BACK
	0	700	JUN	START
	0	010		(address of START)
	0	011		

Here, the first locations of page 7 BANK 0 must be reserved to "recognize" the switch back and to get to the correct location within Bank 0.

The basic programming steps for the switch are as follows:

```
FFB SWITCH,    FIM 0, 0     ;RAM PORT 0 selected
FFD            SRC 0
FFE          * LDM 1        ;Load 0001 to ACC
FFF            WMP
```

* LDM 1 for BANK 1, LDM 0 for BANK 0

The SWITCH may be called from any point *within* the ROM bank in which that SWITCH is located. For example:

BANK	ADDRESS		PROGRAM EXECUTION
0	7A2		:
0	7A3		:
0	7A4		JUN SWITCH
0	FFB	SWITCH	FIM OP 0
0	FFD		SRC OP
0	FFE		* LDM 1
0	FFF		WMP
1	000		:
:	:		:
:	:		:

The switching instructions as given can also be written as a subroutine with a BBL at the end. The program will switch to BANK 1 (for example) and execute from the address just following the address of the JMS (in Bank 0, for example) which called the switch.

When going from bank 0 to bank 1, the next instruction can be a resumption of program flow just prior to the switch, if the normal program would "overflow" into BANK 1. When going from BANK 1 to BANK 0, a determination must be made of where the program should resume executing from.

The first three instructions (FIM, SRC, LDM) can actually be anywhere in the program within one bank, and a JUN FFF will cause the program to switch at the end of ROM if the instruction at FFF is WMP.

If the RAM port which is used for ROM Bank Selection is not used for anything else, the remaining three lines may be used to indicate where the program should go to within the newly selected bank. This is done by testing the accumulator contents *after* the switch. For example:

BANK	ADDRESS		PROGRAM EXECUTION	
	:			
0	FFB	Switch	FIM 0, 0	
0	FFD		SRC 0	
0	FFE		** LDM N	
0	FFF		WMP	
1	000		CLC	
1	001		RAL	
1	002		JC SOMEWHERE	;Simple
	:		RAL	;3 line
:	:		JC SOMEPLACE	;test &
:	:		RAL	;branch
:	:		JC ELSEWHERE	

** LDM N where
N=DDDb

DDD=one of three branch conditions
B=Bank select code

Or, for more branch conditions, test the three DDD bits for 1 of 8 branch codes.

General Notes:

1. The use of the RAM port is advisable since addressing that port is possible from either bank, while ROM ports can only be addressed within the bank with which they are associated. It follows that the ROM port structures associated with bank 0 can be duplicated for bank 1 and used for entirely different purposes. This general scheme (RAM port addressing) can also be used as a means of ROM input and output port expansion for systems using the 4289 or 4008/4009 Standard Memory Interfaces.
2. It is advisable to keep all calls to subroutines within one bank. When switching ROM banks, no subroutine calls should be in effect, i.e., the stack for return addresses is empty. Pseudo-calls to subroutines can be made between banks by using JUN commands to the switch points in each bank and by testing the accumulator to determine which subroutine to go to. The return is a BBL to another

JUN command which switches you back to the previous bank. The use of the JMS and BBL after the switch has been made still leaves the stack empty when the return switch is made, and conforms with normal programming procedures involving subroutines.

3. The RAM port used should be one controlled by CM-RAM 0 since this group of RAM's is automatically selected after power on and reset, thereby assuring that the program begins in ROM bank 0.

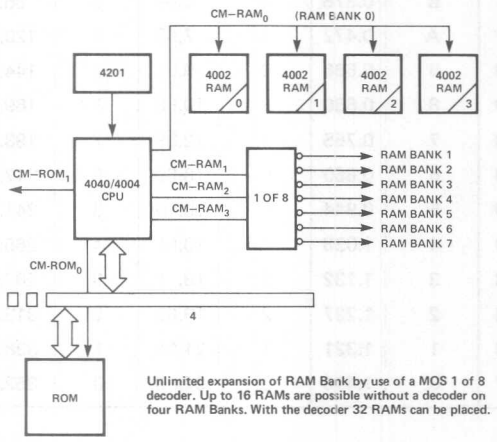


Figure 2-4. CM-RAM Expansion.

Programmed Delays

An operation frequently required in microcomputer system programming is to provide a fixed time delay. A programmed delay can be provided in an MCS-40 system with a series of cascaded ISZ (Increment Index Register and Skip if Zero) instructions operating on a series of index registers with preloaded values. The instructions for the delay would be as follows:

```
      LDM   R1, N1
       :      :        Where Ni is a pre-established
       :      :        value
      LDM   RN, Nn
DLY:  ISZ   R1, DLY
       :      :
       :      :
      ISZ   Rn, DLY
```

NOTE: FIM instructions can also be used to load register pairs.

Table 2-1 provides the values to be loaded into the registers R_i to give a specified total time delay. The table can be used for 1 to 6 cascaded ISZ instructions. The total time delay is given by adding up the times for each ISZ/register count used.

For example, using Table 2-1 to derive a 4 second delay gives the instruction sequence:

```
      FIM   0, 7EH
      FIM   2, 95H
      LDM   4, ODH
DLY4: ISZ   DLY4, 0
      ISZ   DLY4, 1      TTOTAL = 4 = 3.02 +.944 +
      ISZ   DLY4, 2      .0354 +.000389 +.000194
      ISZ   DLY4, 3
      ISZ   DLY4, 4
       :                 ;End of delay
```

Table 2-1 Delay Times Using Cascaded ISZ Instructions
(Assumes 10.8 μs Instruction Execution Cycle)
$T_{TOTAL} = T_0 + T_1 + T_2 + T_3 + T_4 + T_5$

Number of Counts	Reg. 0 Value	T_0 Micro Sec.	Reg. 1 Value	T_1 Milli Sec.	Reg. 2 Value	T_2 Milli Sec.	Reg. 3 Value	T_3 Sec.	Reg. 4 Value	T_4 Sec.	Reg. 5 Value	T_5 Sec.
1	F	21.6	F	0.022	F	0.02	F	0.000	F	0.00	F	0.0
2	E	43.2	E	0.389	E	5.92	E	0.094	E	1.51	E	24.2
3	D	64.8	D	0.756	D	11.82	D	0.189	D	3.02	D	48.3
4	C	86.4	C	1.123	C	17.71	C	0.283	C	4.53	C	72.4
5	B	108.0	B	1.490	B	23.61	B	0.378	B	6.04	B	96.6
6	A	129.6	A	1.858	A	29.51	A	0.472	A	7.55	A	120.7
7	9	151.2	9	2.225	9	35.40	9	0.566	9	9.06	9	144.9
8	8	172.8	8	2.592	8	41.30	8	0.660	8	10.56	8	169.0
9	7	194.4	7	2.959	7	47.20	7	0.755	7	12.08	7	193.2
10	6	216.0	6	3.326	6	53.09	6	0.850	6	13.59	6	217.3
11	5	237.6	5	3.694	5	58.99	5	0.944	5	15.10	5	241.5
12	4	259.2	4	4.061	4	64.89	4	1.038	4	16.61	4	265.6
13	3	280.8	3	4.317	3	70.78	3	1.132	3	18.12	3	289.8
14	2	302.4	2	4.795	2	76.68	2	1.227	2	19.63	2	313.9
15	1	224.0	1	5.162	1	82.58	1	1.321	1	21.14	1	338.1
16	0	345.6	0	5.530	0	88.47	0	1.416	0	22.65	0	362.2

Use of Designate ROM Bank Instructions

The DB instructions present a convenient method of switching from one ROM bank to another. As shown in the following examples, the bank switch is delayed until the 3rd instruction cycle after the DB is executed.

Example #1:

ROM Location* Bank / Page / Word	Instruction	Comment
0 / 2 / 107	X X X	
0 / 2 / 108	DB1	Designate Bank 1.
0 / 2 / 109	JUN 1	
0 / 2 / 110	27	During this instruction cycle a "1" is loaded in bit #3 of the command register.
1 / 1 / 27	X X X	JUN occurred to Bank 1 because CM-ROM$_1$ has been activated.
.	.	
.	.	
.	.	
1 / 1 / 63	DB0	Designate Bank 0.
1 / 1 / 64	ISZ 3	
1 / 1 / 65	151	
0 / 1 / 66	X X X	Program jumps here if (IR_3) = 0.
0 / 1 / 151	X X X	Program jumps here if (IR_3) ≠ 0.

Example #2

ROM Location Bank / Page / Word	Instruction	Comment
0 / 7 / 131	X X X	
0 / 7 / 132	DB1	Designate Bank 1.
0 / 7 / 133	JMS 2	
0 / 7 / 134	96	Address 7, 135 saved in stack.
1 / 2 / 96	X X X	JMS occurs to 1, 2, 96 since CM-ROM, is activated at this instruction cycle.
.	.	
.	.	
.	.	
1 / 2 / 170	DB0	Designate Bank 0.
1 / 2 / 171	XCH 7	
1 / 2 / 172	BBL	Address 7, 135 pulled from stack and placed in PC; branch back occurs to 7, 135 in Bank 0 because CM-ROM$_0$ is activated during this instruction cycle.
0 / 7 / 135	X X X	

* Bank # 0, 1 Page # 0 - 15 Word # 0 - 255

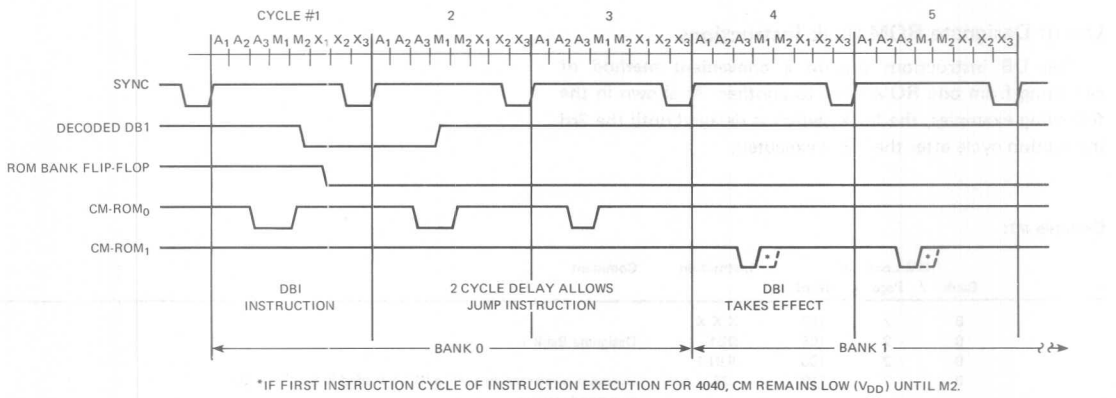


Figure 2-5. ROM Bank Switching.

Interrupt Processing

The Interrupt signal, when armed and activated, causes the CPU to suspend normal program execution. The CPU is forced to execute a predetermined Interrupt subroutine starting from location 003. At the completion of the Interrupt routine, the CPU is returned to the normal program execution with a BBS instruction.

The Interrupt utilizes the address push down stack. When an Interrupt is executed, the program counter is pushed up one level and is reloaded with address 003. Execution will proceed from this location. The location may contain Jump (JIN, JUN) which allows the user to place the Interrupt routine anywhere in memory. A stack level should always be reserved for the interrupts to avoid overflowing the stack.

Since the Interrupt forces the CPU out of the normal instruction sequence, the state of the CPU's internal register values must be preserved and restored prior to returning from the Interrupt routine. The SRC and the Index Register Bank will automatically be saved. The designer should store the value of the *accumulator* and *carry flip-flop, command register, current ROM bank*, and *index registers* that will be used during the Interrupt execution.

The 4040 CPU has three groups of eight index registers organized into two banks. Registers 8-15 are common to both. Bank 0, Register 0-7 can be designated for normal program execution while Bank 1, Register 0-7 can be designated for interrupt execution. The designer need only switch banks to save the first eight register values. These will be restored automatically with the BBS.

The programmer must first enable the CPU to accept an Interrupt. This is done by executing the EIN instruction. When an Interrupt occurs, the program counter is forced to location 003 in whichever ROM bank it is executing.

An interrupt processing routine is, in general, composed of three distinct parts:

a. The instructions required to save the current processor status.

b. A portion which determines and services the interrupting device.

c. The instructions required to restore program control to the pre-interrupt conditions.

In the following example, the processor is used with a single ROM bank, and Index Register (IR) Bank 1 is used to save status (accumulator/carry, Command Register (CR)). The six remaining registers in IR Bank 1 are available for interrupt servicing. In addition to being relatively simple, this scheme has the advantage of saving processor status with the fewest number of instructions. Note that since only one ROM bank is available, it is only necessary to save the lower three bits of CR. This allows saving the CR and CY to be merged in the same register location.

Example #3:

ROM Location Bank / Page / Word			Instruction	Comment
0	6	82	SRC 4	(IR 8,9) sent to ROM & RAM, Load SRC Reg.
0	6	83	INC 9	Interrupt occurs here.
0	6	84	(JMS 0)	Interrupt acknowledged, 6,84 saved in stack; instruction at 6,84 ignored.
0	6	84	(3)	
0	0	3	SB1	Select IR Bank 1.
0	0	4	XCH 7	(ACC) → IR7 – ACC saved.
0	0	5	LCR	(CR) → ACC
0	0	6	RAL	(CY) → Acc_0, Acc_0 → Acc_1 . . . Acc_3 → CY
0	0	7	XCH 6	(ACC) → IR6 CY, CR saved.
0	0	8		
0	0	9	.	Routine for determining and servicing interrupt is executed here.
	.		.	
	.		.	
	.		.	
0	P	n	XCH 6	(IR6) → ACC
		n+1	RAR	ACC_0 → CY – CY restored
		n+2	DCL	ACC_0 → CR_0, ACC_1 → CR_1, ACC_2 → CR_2, CR restored.
		n+3	XCH 7	(IR7) → ACC
		n+4	BBS	Address 6,84 loaded into PC; contents of SRC register sent out and the Index Register Bank selection is restored. The Interrupt Acknowledge line is cleared (to V_{SS}).
0	6	84	WRM	Program restored.
			.	
			.	
			.	

*Index Register Bank is automatically restored with BBS.

CHAPTER 3
INTERFACE DESIGN WITH THE MCS-40™ SYSTEM

INTERFACE DESIGN TECHNIQUES

MCS-40 computer systems are often used to replace random logic controllers in a wide variety of systems. In each of these systems a number of peripheral devices, such as keyboards, switches, indicator lamps, numeral displays, printer mechanisms, relays, solenoids, etc., may have to be interrogated or controlled. The engineer who wishes to utilize an MCS-40 system must include, as part of his design, suitable interface circuits and programs.

Devices to be operated or interrogated by an MCS-40 computer are attached to the system via the input and output data ports. The design of an interface consists of the following steps:

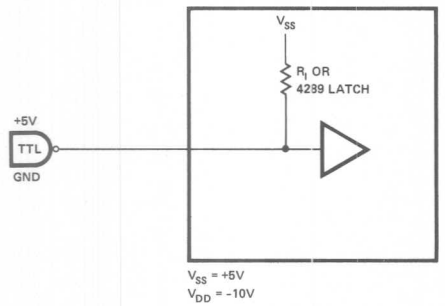
1. Establish I/O configuration by assigning peripheral device connections to port connections. If the number of available output ports is insufficient, General Purpose I/O devices such as the 4265, may be added or output port expanders, such as the 4003, may be used. Reducing the number of I/O lines can be accomplished by the use of multiplexers. These multiplexers can be controlled by output ports.
2. Develop the necessary level conditioning circuits for each signal. For TTL and CMOS compatibility refer to the Electrical Interfacing section which follows.
3. Write the programs necessary to interpret inputs and generate the output levels necessary for proper operation of the peripherals. This is best done by use of a functional flow diagram.

Any interface design requires all three of these steps. Each design will typically involve decisions concerning the interaction of the three areas.

ELECTRICAL INTERFACING

Interfaces for: 4001
4008/4009
4289
4308

1. TTL to MCS-40 Input Port

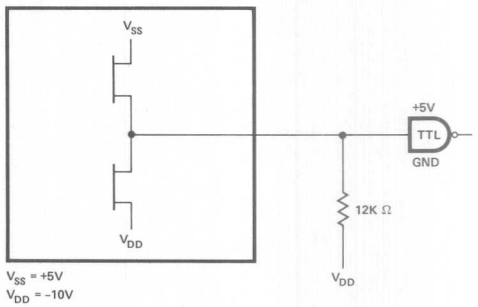


Caution, if R_I not connected to V_{SS}, then use external 5.1KΩ resistor to V_{CC} to meet MCS-40 V_{IH}.

NOTES:

1. The input options for the ROMs are shown in Chapter 5 with the detailed description of the ROM I/O ports.
2. The 4001 non-inverting input is not allowed for TTL to 4001 input.

2. MCS-40 Output Port to TTL



Use 12K Ω for 4001, 4002, 4308 to standard TTL.

Omit 12K Ω for 4009 or 4289, or to Low Power TTL or Schottky TTL with PNP inputs, where $I_F \leq 0.8$mA.

Use of 4308 R_I to V_{DD} allows higher than 12K Ω external resistor, but does not remove need for resistor.

---

3. MCS-40 Output Port to CMOS

Case 1:

MCS-40: V_{SS} = +5V, V_{DD} = -10V, V_{DD1} = GND

CMOS: V_{SS} = GND, V_{DD} = +5V

V_{IH} and V_{IL} to CMOS: No problem.

Input low rating to CMOS: The MCS-40 output must be clamped to CMOS GND by a resistor-diode network or by a diode alone to limit the CMOS input current below ground.

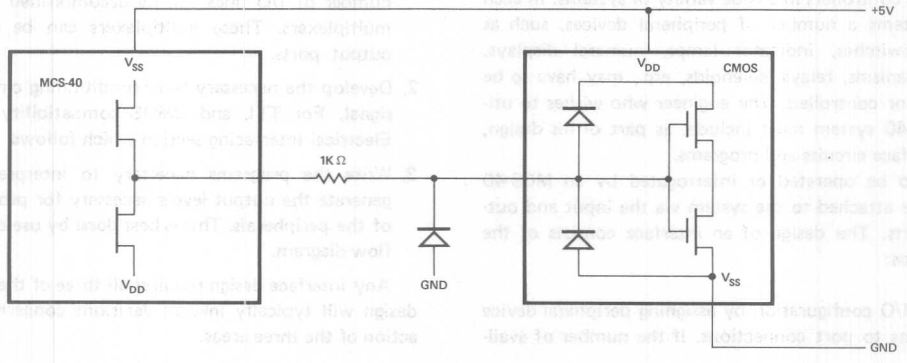
Case 2:

MCS-40: V_{SS} = +5V

CMOS: V_{SS} = -5 to -10V, V_{DD} = +5V

V_{IH} to CMOS: No problem.

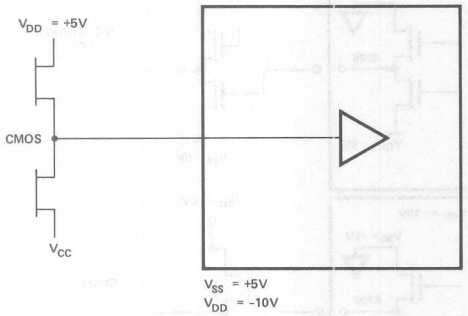
V_{IL} to CMOS: Input low to CMOS — Use 50K Ω resistor to CMOS V_{SS}



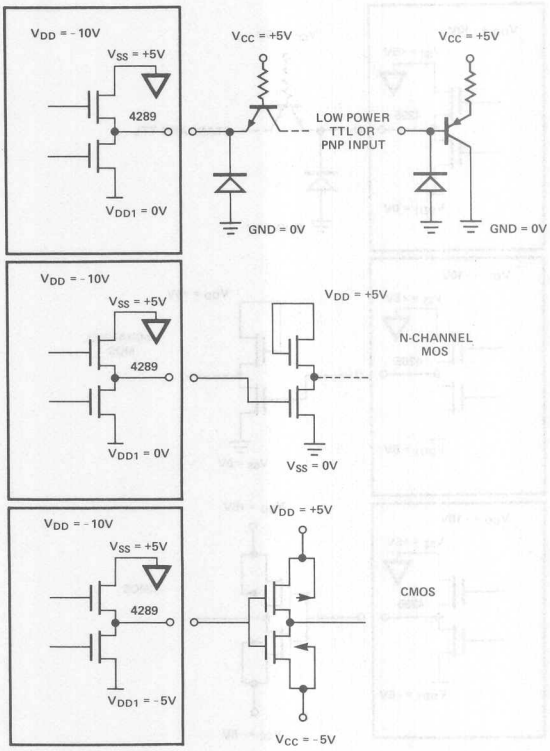
4. CMOS to MCS-40 Input Port

No problem for CMOS $V_{DD} \geqslant 5V$ and CMOS V_{DD} (positive rail) = MCS-40 V_{SS} for 4001, 4009, 4308.

For 4289, to overcome I/O latches, CMOS low level $Z_0 \leqslant 1.5K\ \Omega$ to V_{SS} -5V.

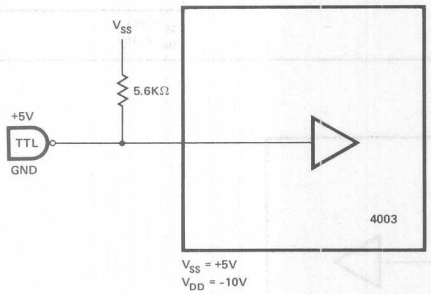


5. 4289 Interface Method Using V_{DD1}

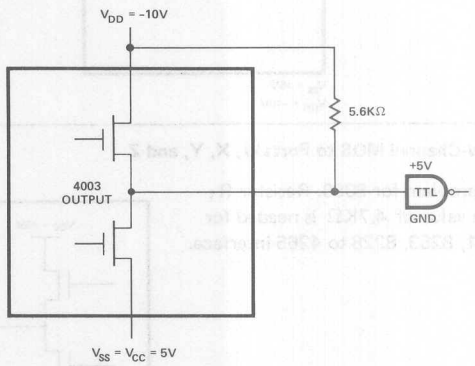


Interfaces for 4003

1. TTL to 4003 Input Pin

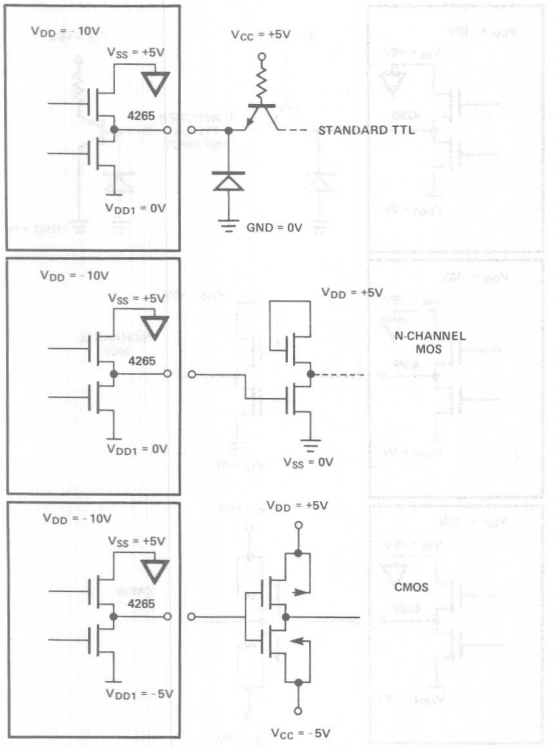


2. 4003 Output to TTL

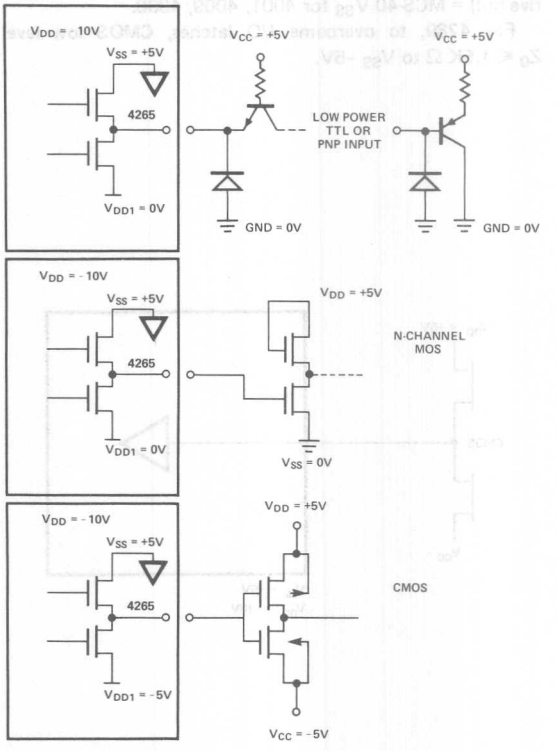


Interfaces for 4265

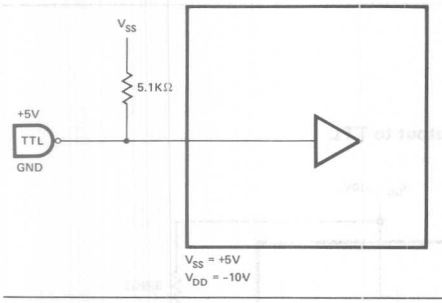
1. Port Z Output Using V_{DD1}



2. Ports W, X, and Y Output Using V_{DD1}

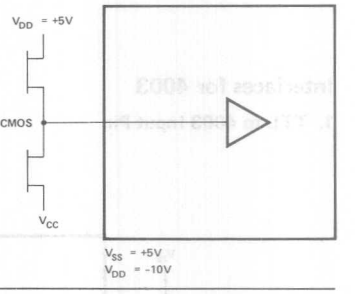


3. TTL Input to Ports W, X, Y, and Z



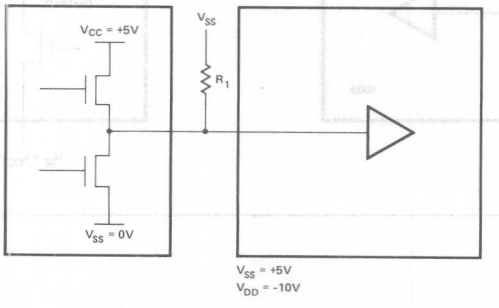
4. CMOS to Ports W, X, Y, and Z

No problem for CMOS $V_{DD} \geq 5V$ and CMOS V_{DD} (positive rail) = MCS-40 V_{SS}.



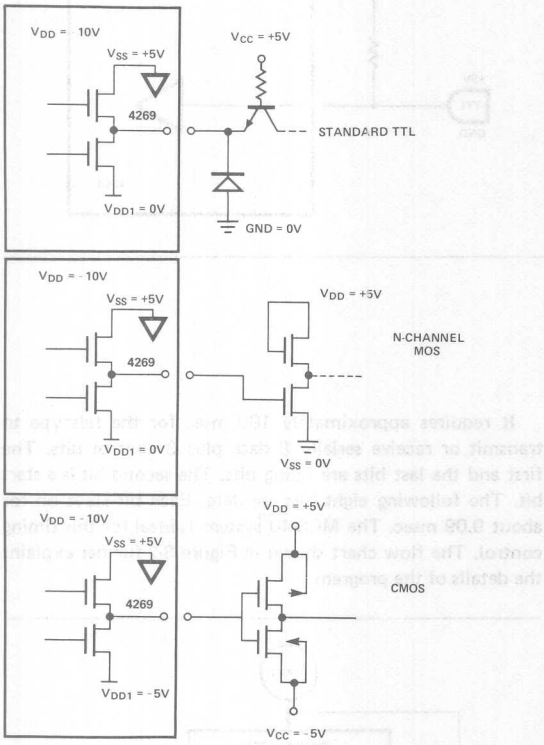
5. N-Channel MOS to Ports W, X, Y, and Z

No problem for 8080. Resistor R_1 with value of 4.7KΩ is needed for 8251, 8253, 8228 to 4265 interface.

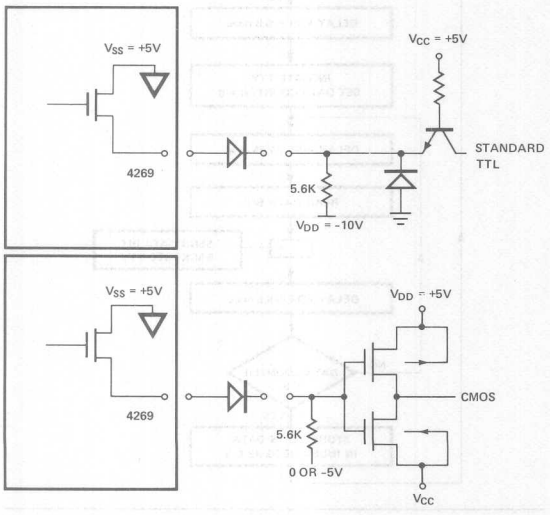


Interfaces for 4269

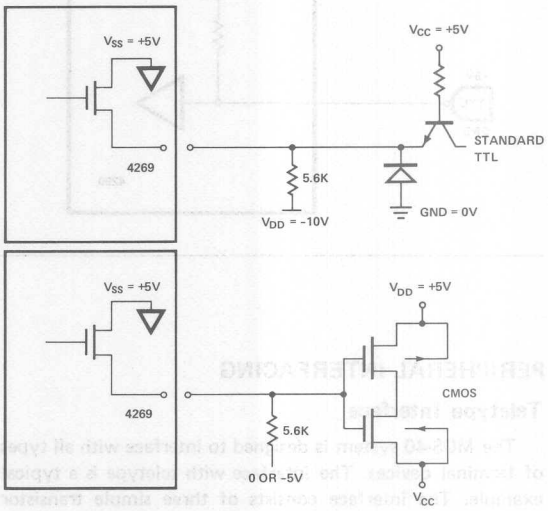
1. A and B Display Outputs



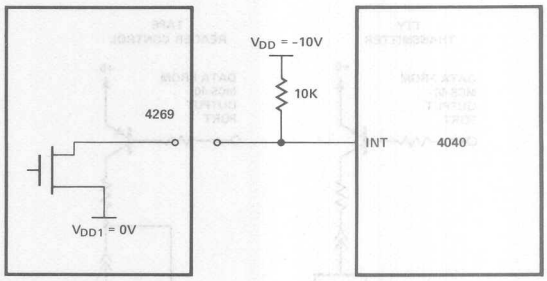
2. Scan Outputs (S_0 - S_7)



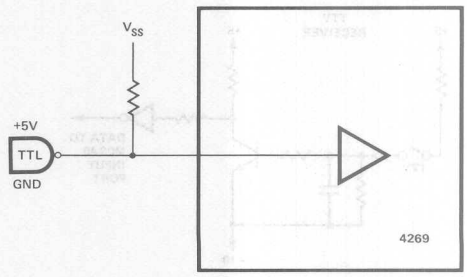
3. RS Output



4. INT Output

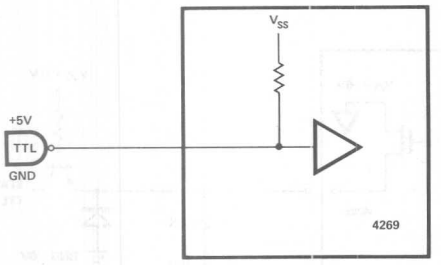


5. Shift and Strobe/Control (S/C) Inputs

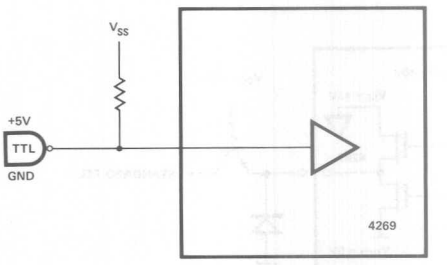


6. Return Lines (R_0 - R_7)

a. Scanned or Encoded Keyboard Mode



b. Sensor Mode



PERIPHERAL INTERFACING

Teletype Interface

The MCS-40 system is designed to interface with all types of terminal devices. The interface with teletype is a typical example. The interface consists of three simple transistor circuits. (See Figure 3-1.) One transistor is used for receiving serial data from the teletype, one for transmitting data back into the teletype, and the third one for tape reader control.

It requires approximately 100 msec for the teletype to transmit or receive serially 8 data plus 3 control bits. The first and the last bits are idling bits. The second bit is a start bit. The following eight bits are data. Each bit stays on for about 9.09 msec. The MCS-40™ system is ideal for this timing control. The flow chart shown in Figure 3-2 further explains the details of the program.

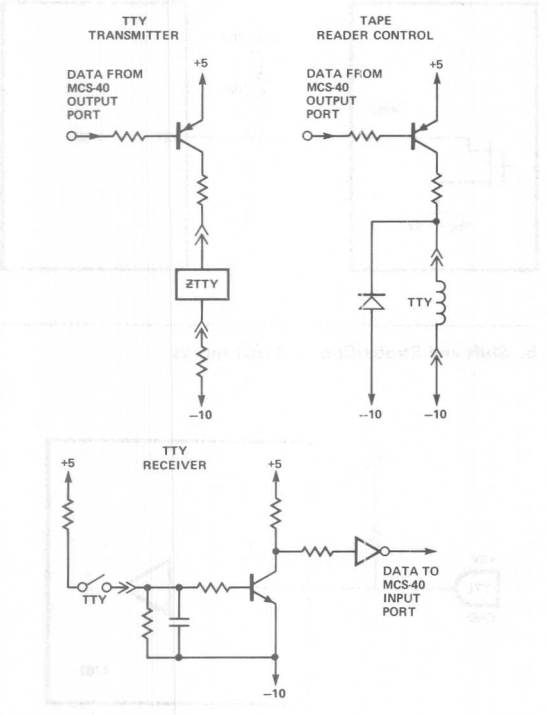


Figure 3-1. MCS-40 and Teletype Interface Circuits.

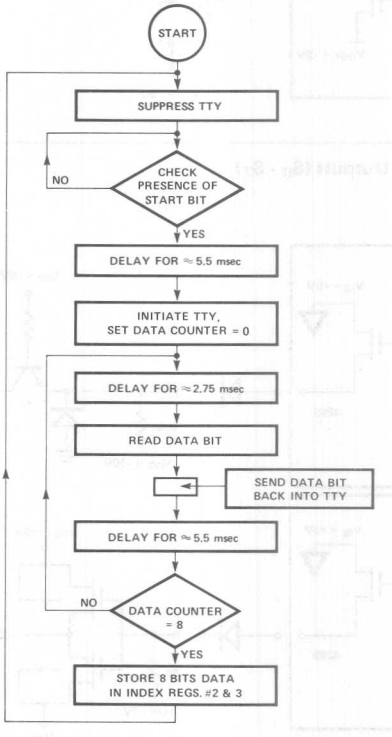


Figure 3-2. Flow Chart for Teletype Interfaces.

Keyboards

The MCS-40 microcomputer can scan and debounce a keyboard or can interface to a keyboard which presents precoded data (such as ASCII). The keyboard portion of the 4269 Programmable Keyboard/Display device is especially designed to perform these functions. Chapter 5 describes the 4269 keyboard operations in detail.

Display

Display devices such as Self-Scan® displays and LED arrays are easily interfaced to the MCS-40 system. The display portion of the 4269 Programmable Keyboard/Display is designed specifically for interfacing to such devices.

Printer

Most standard printer interfaces are parallel in nature. The interface signals may be divided into two groups, data and control. The control section would typically consist of forms manipulation and printing handshaking. Due to their parallel nature, a general purpose I/O device such as the 4265 lends itself well to this application. Sixteen bits of I/O and printer functions can be assigned to the various bit lines and programmed accordingly. An individual bit/reset function such as is found in the 4265 is especially powerful in printer applications.

Other Peripherals

An MCS-40 system can interface to many widely varying peripheral devices besides the ones described above. Refer to Chapter 4 for several typical system applications illustrating examples of different peripheral interfaces.

NOTE: Self-Scan is a registered trademark of Burroughs Corporation.

CHAPTER 4
SYSTEM CONFIGURATIONS

SYSTEM CONFIGURATIONS

The following diagrams illustrate various configurations in which the MCS-40™ components can be applied. All of the block diagrams can be expanded to accommodate a specific application.

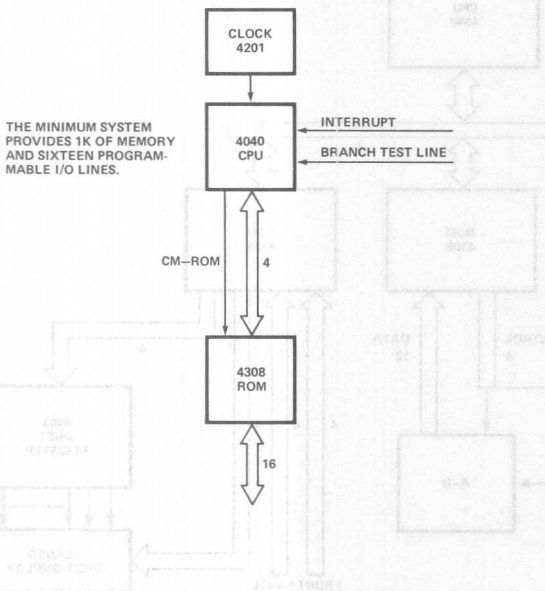


Figure 4-1. The Minimum System Provides 1K of Memory and Sixteen Programmable I/O Lines.

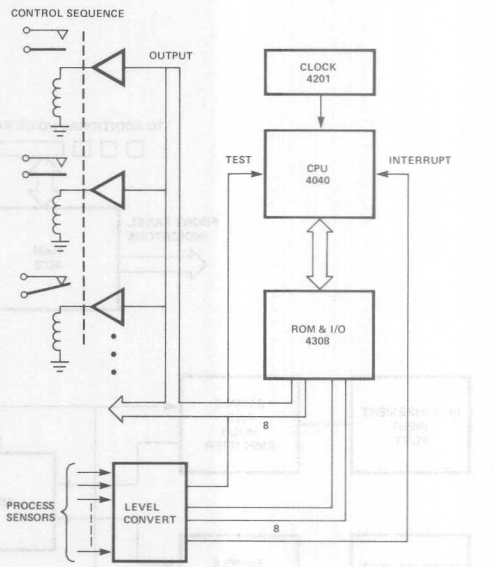


Figure 4-2. Basic Process Controller.

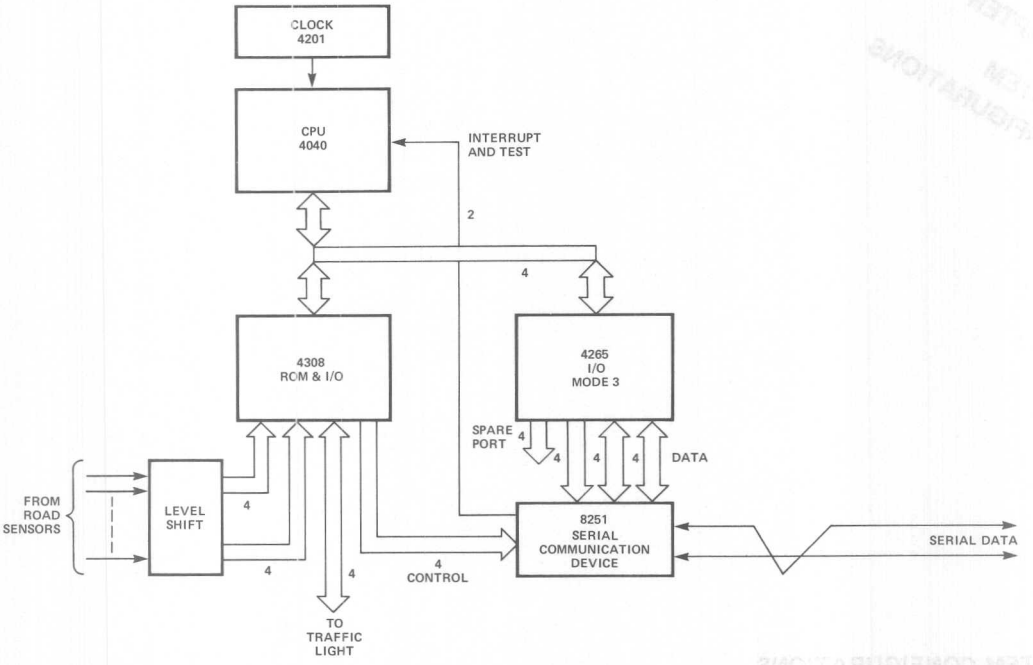


Figure 4-3. Traffic Light Control.

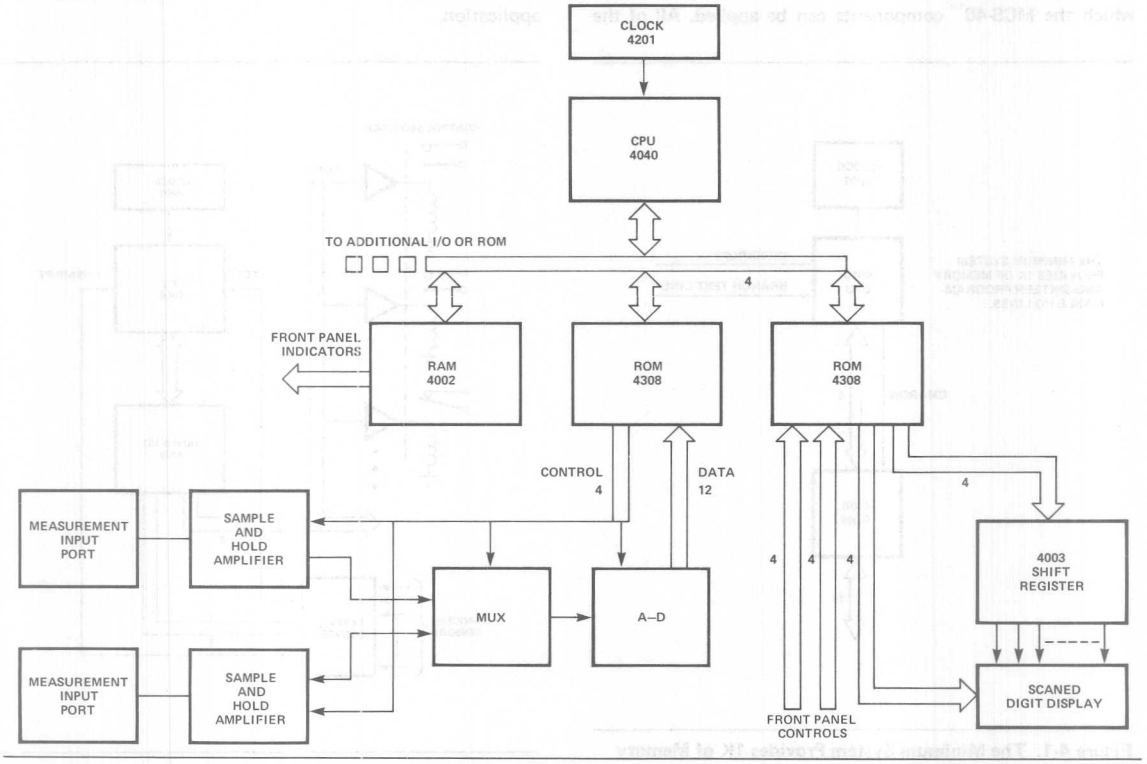


Figure 4-4. Typical A-D Instrument.

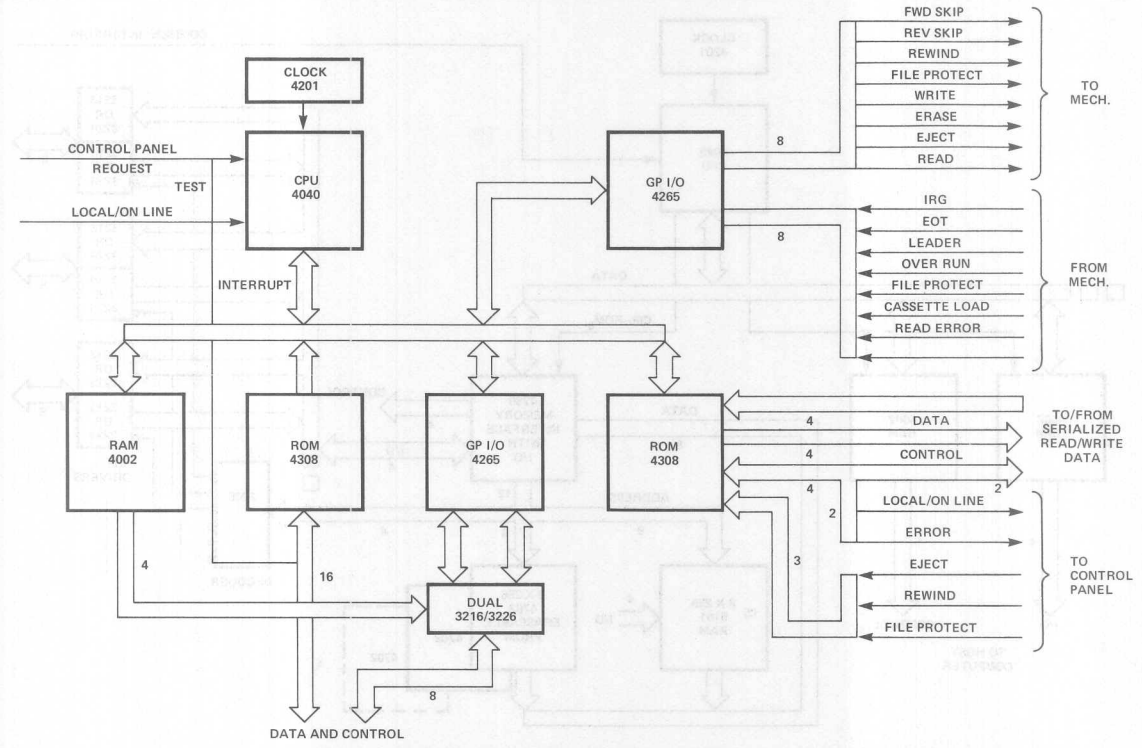


Figure 4-5. Cassette Controller.

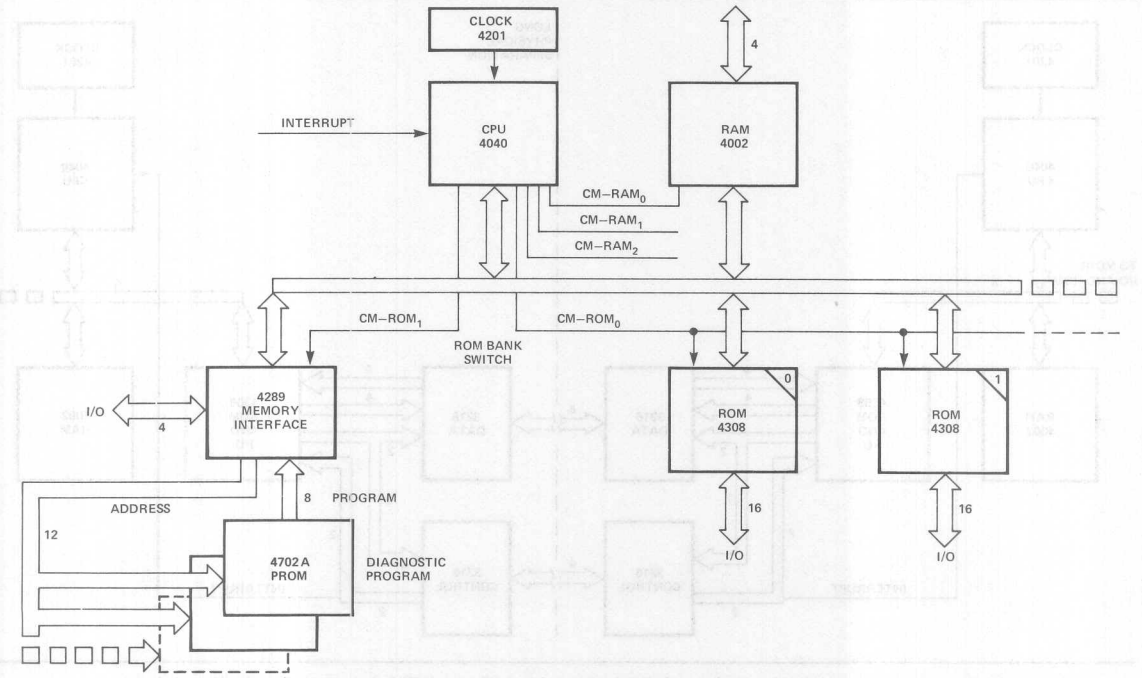


Figure 4-6. System Diagram Indicating PROM/ROM Memory Environment and the Use of the 4289 and 4702.

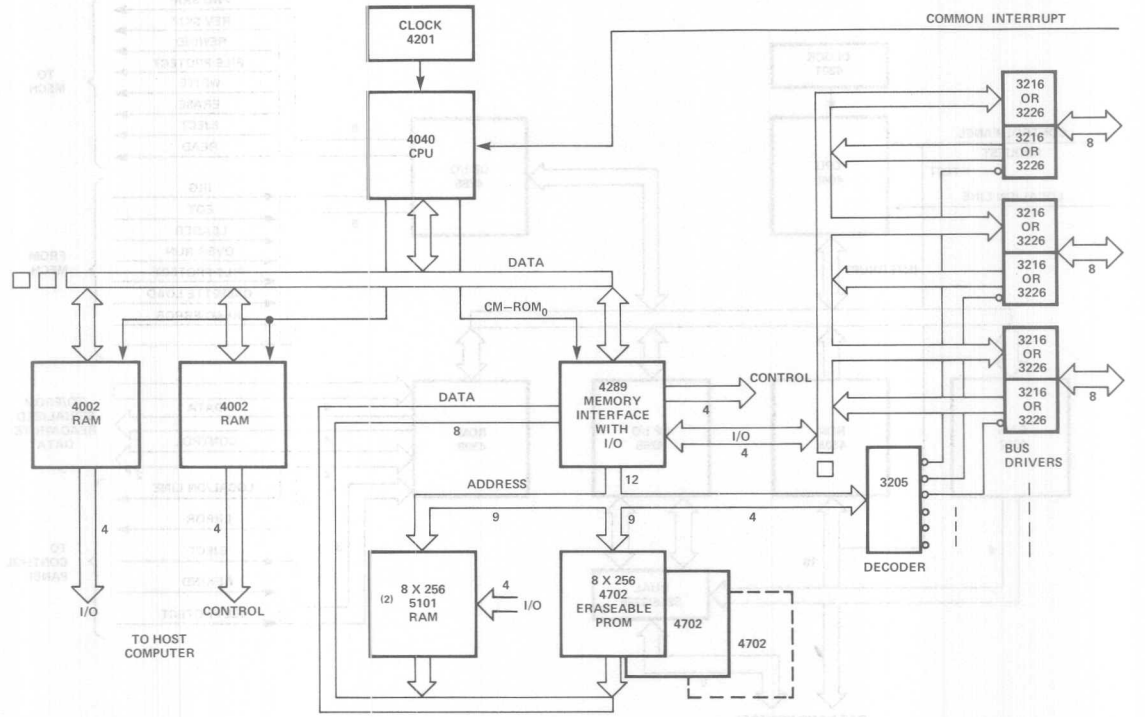


Figure 4-7. Multiple 8-Bit I/O Port with Writeable Program Memory.

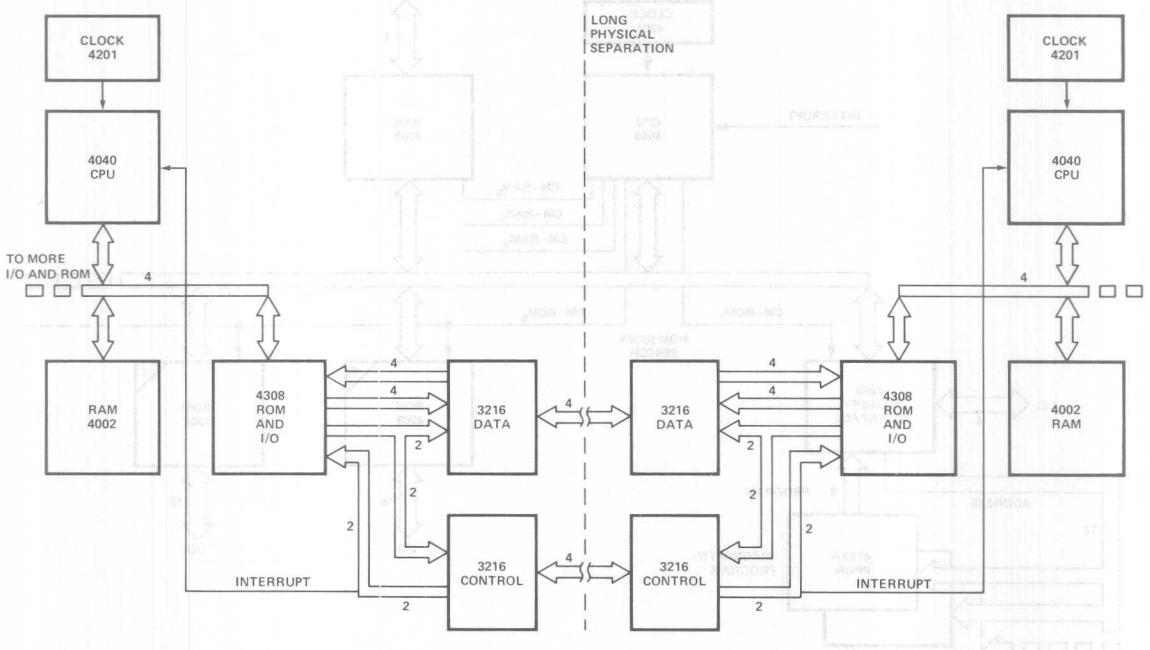


Figure 4-8. Dual Processor Communications Allows for Distributed Intelligence. This Interface Can Be Expanded to Provide any Width Word.

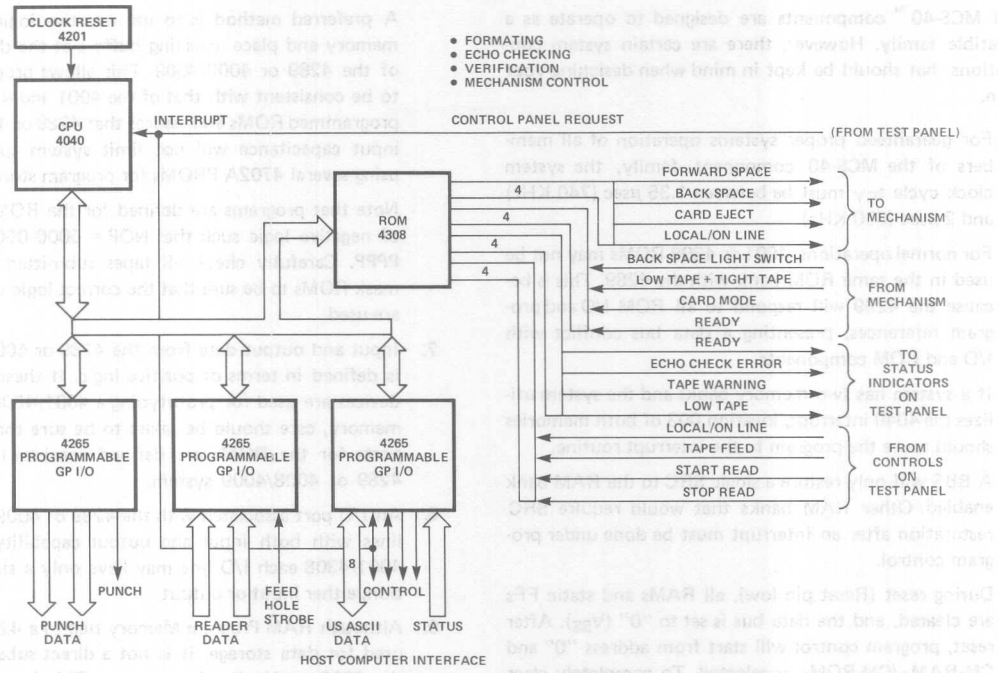


Figure 4-9. Paper Tape Reader and Punch Controller.

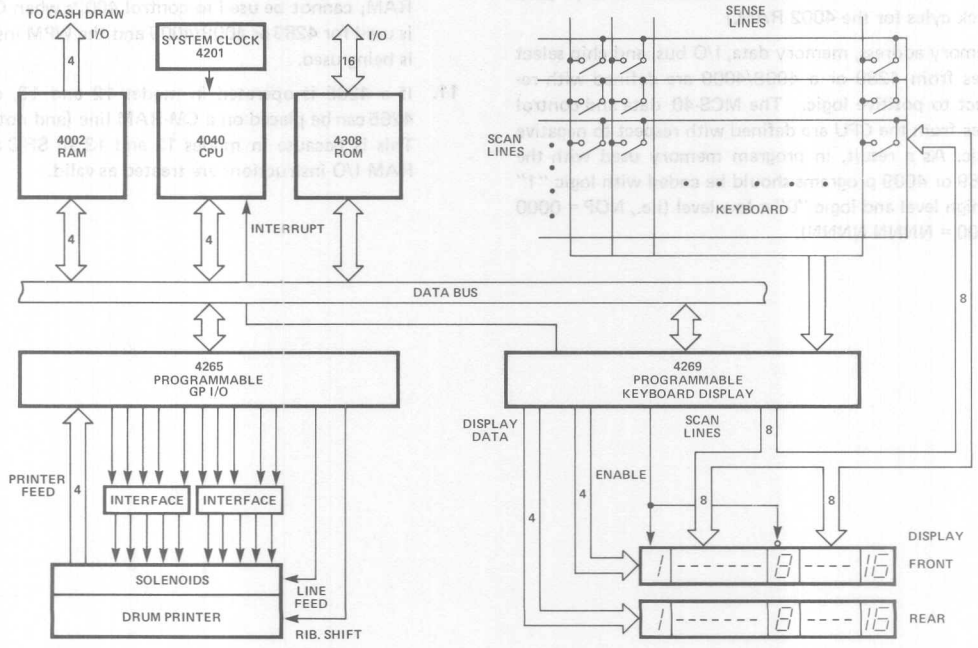


Figure 4-10. Basic Printer/Display Terminal.

SYSTEM CONSIDERATIONS

All MCS-40™ components are designed to operate as a compatible family. However, there are certain system considerations that should be kept in mind when designing your system.

1. For guaranteed proper systems operation of all members of the MCS-40 component family, the system clock cycle t_{CY} must be between 1.35 μsec (740 KHz) and 2 μsec (500 KHz).
2. For normal operations, 4001 or 4308 ROMs may not be used in the same ROM bank with the 4289. This is because the 4289 will respond to all ROM I/O and program references, presenting a data bus conflict with I/O and ROM components.
3. If a system has two memory banks and the system utilizes the 4040 interrupt, location 003 of both memories should route the program to the interrupt routine.
4. A BBS will only restore a single SRC to the RAM bank enabled. Other RAM banks that would require SRC restoration after an Interrupt must be done under program control.
5. During reset (Reset pin low), all RAMs and static FFs are cleared, and the data bus is set to "0" (V_{SS}). After reset, program control will start from address "0" and CM-RAM_0/CM-ROM_0 is selected. To completely clear all registers and RAM locations in the CPU, the reset signal must be applied for at least 12 full instruction cycles (96 clock cycles) to allow the index register refresh counter to scan all locations in memory. (256 clock cyles for the 4002 RAM.)
6. Memory address, memory data, I/O bus, and chip select lines from 4289 or a 4008/4009 are defined with respect to positive logic. The MCS-40 data and control lines from the CPU are defined with respect to negative logic. As a result, in program memory used with the 4289 or 4009 programs should be coded with logic "1" = high level and logic "0" = low level (i.e., NOP = 0000 0000 = NNNN NNNN).

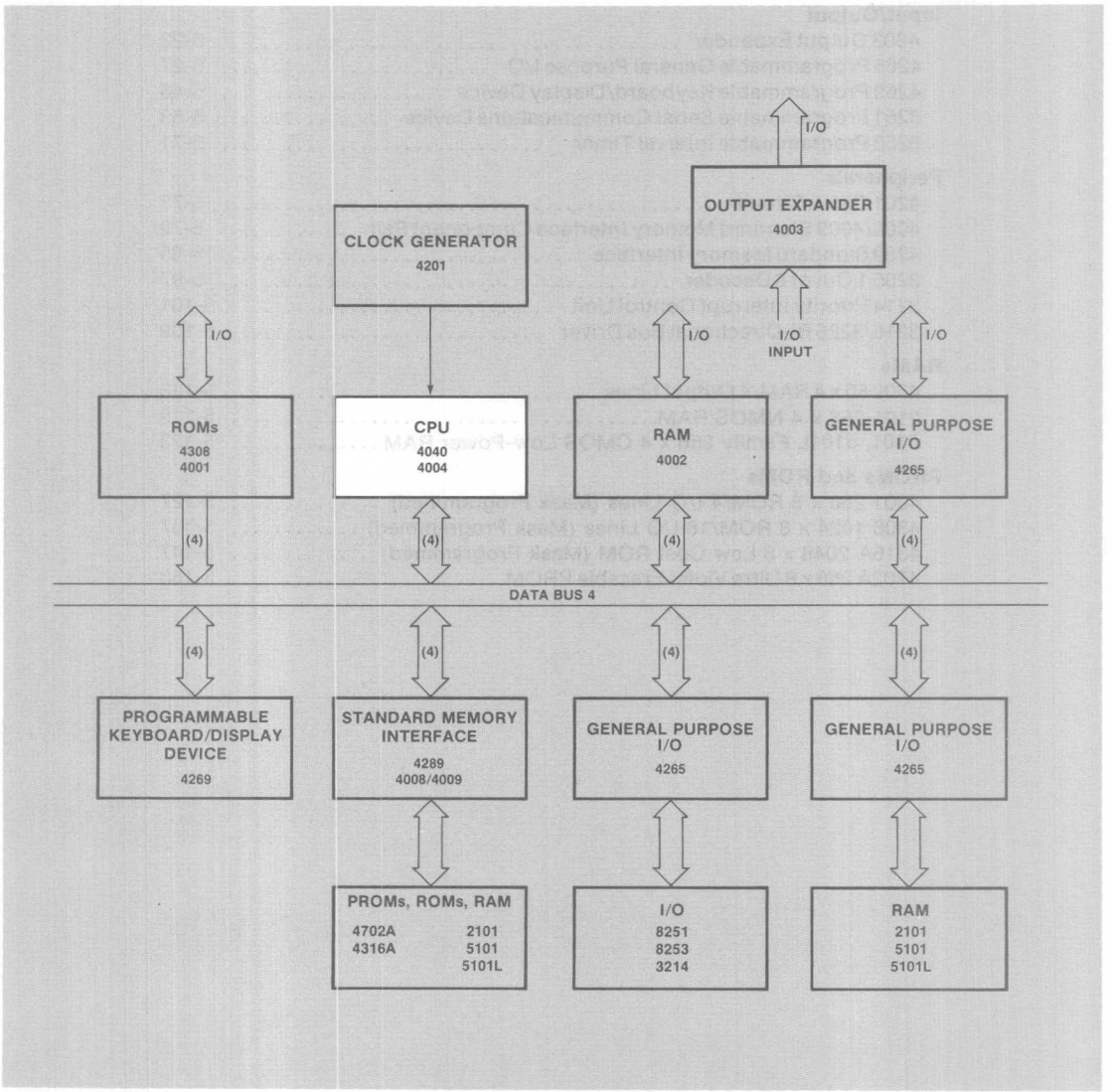
   A preferred method is to use negative logic program memory and place inverting buffers at the data inputs of the 4289 or 4008/4009. This allows program code to be consistent with that of the 4001 and 4308 mask-programmed ROMs and assures that 4289 or 4008/4009 input capacitance will not limit system speed when using several 4702A PROMs for program storage.

   Note that programs are defined for the ROM in terms of negative logic such that NOP = 0000 0000 = PPPP PPPP. Carefully check all tapes submitted for metal mask ROMs to be sure that the correct logic definitions are used.
7. Input and output data from the 4289 or 4009 I/O bus is defined in terms of positive logic. If these interface devices are used for prototyping a 4001/4308 program memory, care should be taken to be sure that the I/O ports for the ROMs are defined consistent with the 4289 or 4008/4009 system.
8. An I/O port associated with the 4289 or 4009 can have lines with both input and output capability. On the 4001/4308 each I/O line may have only a single function, either input or output.
9. Although RAM Program Memory tied to a 4289 can be used for data storage, it is not a direct substitute for the 4002 read/write data storage. This is because the data RAM instructions such as SBM and ADM address 4002s, not program memory.
10. For 4002s with FPO numbers lower than H7224, CM-RAM_i cannot be used to control 4002s when CM-ROM is used for 4289 or 4008/4009 and the WPM instruction is being used.
11. If a 4265 is operated in modes 12 and 13, only the 4265 can be placed on a CM-RAM line (and not 4002s). This is because in modes 12 and 13 all SRC and CM-RAM I/O instructions are treated as valid.

CHAPTER 5
MCS-40™ COMPONENT FAMILY

intel Microcomputer Systems

CPUs
4004
4040



4004

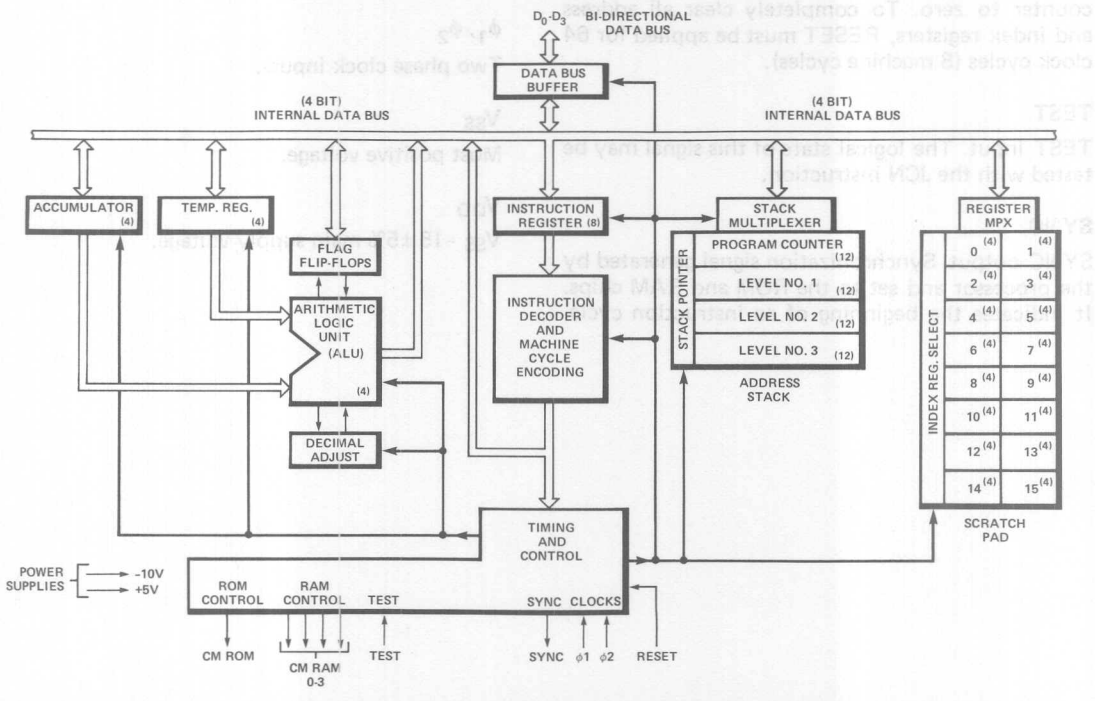
SINGLE CHIP 4-BIT P-CHANNEL MICROPROCESSOR

- 4-Bit Parallel CPU With 46 Instructions
- Instruction Set Includes Conditional Branching, Jump to Subroutine and Indirect Fetching
- Binary and Decimal Arithmetic Modes
- 10.8 Microsecond Instruction Cycle
- CPU Directly Compatible With MCS-40 ROMs and RAMs
- Easy Expansion—One CPU can Directly Drive up to 32,768 Bits of ROM and up to 5120 Bits of RAM
- Standard Operating Temperature Range of 0° to 70°C
- Also Available With -40° to +85°C Operating Range

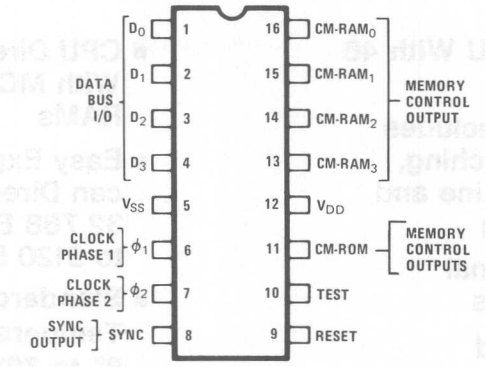
The Intel® 4004 is a complete 4-bit parallel central processing unit (CPU). The 4004 easily interfaces with keyboards, switches, displays, A-D converters, printers and other peripheral equipment.

The CPU can directly address 4K 8-bit instruction words of program memory and 5120 bits of data storage RAM. Sixteen index registers are provided for temporary data storage. Up to 16 4-bit input ports and 16 4-bit output ports may also be directly addressed.

The 4004 is fabricated with P-channel silicon gate MOS technology.



Pin Description



D_0-D_3

BIDIRECTIONAL DATA BUS. All address and data communication between the processor and the RAM and ROM chips occurs on these 4 lines.

RESET

RESET input. A logic "1" level at this input clears all flags and status registers and forces the program counter to zero. To completely clear all address and index registers, RESET must be applied for 64 clock cycles (8 machine cycles).

TEST

TEST input. The logical state of this signal may be tested with the JCN instruction.

SYNC

SYNC output. Synchronization signal generated by the processor and set to the ROM and RAM chips. It indicates the beginning of an instruction cycle.

CM-ROM

CM-ROM output. This is the ROM selection signal sent out by the processor when data is required from program memory.

$CM\text{-}RAM_0$ – $CM\text{-}RAM_3$

CM-RAM outputs. These are the bank selection signals for the 4002 RAM chips in the system.

ϕ_1, ϕ_2

Two phase clock inputs.

V_{SS}

Most positive voltage.

V_{DD}

V_{SS} -15 ±5% main supply voltage.

Instruction Set Format

A. Machine Instructions

- 1 word instruction – 8-bits requiring 8 clock periods (instruction cycle).
- 2 word instruction – 16-bits requiring 16 clock periods (2 instruction cycles).

Each instruction is divided into two four-bit fields. The upper 4-bits is the OPR field containing the operation code. The lower 4-bits is the OPA field containing the modifier. For two word instructions, the second word contains address information or data.

The upper 4-bits (OPR) will always be fetched before the lower 4-bits (OPA) during M_1 and M_2 times respectively.

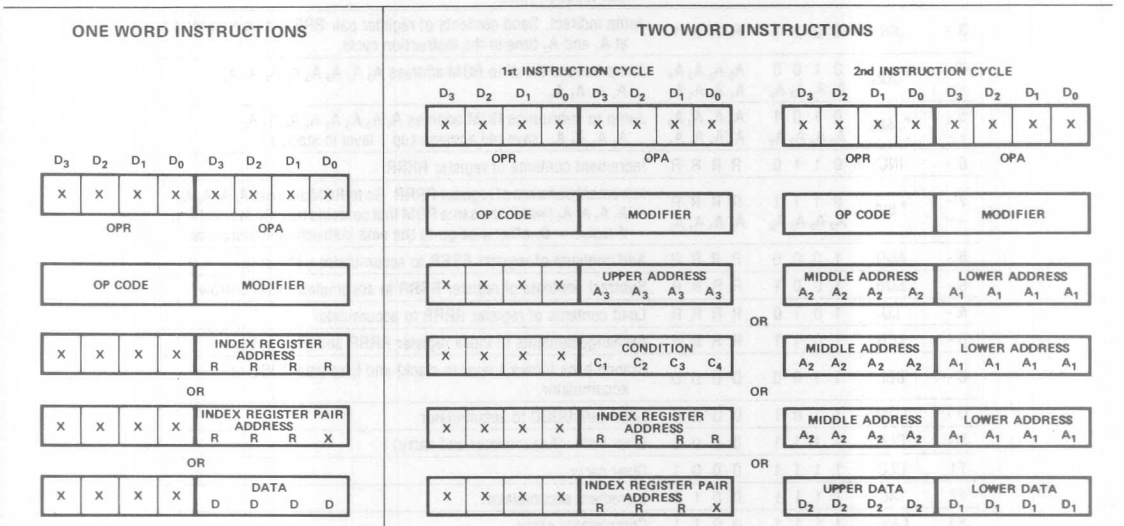


Table I. Machine Instruction Format

B. Input/Output and RAM Instructions and Accumulator Group Instructions

In these instructions (which are all single word) the OPR contains a 4-bit code which identifies either the I/O instruction or the accumulator group instruction and the OPA contains a 4-bit code which identifies the operation to be performed. Table II illustrates the contents of each 4-bit field.

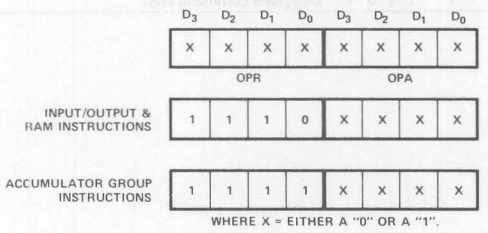


Table II. I/O and Accumulator Group Instruction Formats

4004 Instruction Set

BASIC INSTRUCTIONS (* = 2 Word Instructions)

Hex Code	MNEMONIC	OPR $D_3\ D_2\ D_1\ D_0$	OPA $D_3\ D_2\ D_1\ D_0$	DESCRIPTION OF OPERATION
00	NOP	0 0 0 0	0 0 0 0	No operation.
1 -	*JCN	0 0 0 1	$C_1\ C_2\ C_3\ C_4$	Jump to ROM address $A_2\ A_2\ A_2\ A_2$, $A_1\ A_1\ A_1\ A_1$ (within the same ROM that contains this JCN instruction) if condition $C_1\ C_2\ C_3\ C_4$ is true, otherwise go to the next instruction in sequence.
- -		$A_2\ A_2\ A_2\ A_2$	$A_1\ A_1\ A_1\ A_1$	
2 -	*FIM	0 0 1 0	R R R 0	Fetch immediate (direct) from ROM Data $D_2\ D_2\ D_2\ D_2\ D_1\ D_1\ D_1\ D_1$ to index register pair location RRR.
- -		$D_2\ D_2\ D_2\ D_2$	$D_1\ D_1\ D_1\ D_1$	
3 -	FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
3 -	JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A_1 and A_2 time in the instruction cycle.
4 -	*JUN	0 1 0 0	$A_3\ A_3\ A_3\ A_3$	Jump unconditional to ROM address $A_3\ A_3\ A_3\ A_3\ A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1\ A_1$.
- -		$A_2\ A_2\ A_2\ A_2$	$A_1\ A_1\ A_1\ A_1$	
5 -	*JMS	0 1 0 1	$A_3\ A_3\ A_3\ A_3$	Jump to subroutine ROM address $A_3\ A_3\ A_3\ A_3\ A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1\ A_1$, save old address (up 1 level in stack.)
- -		$A_2\ A_2\ A_2\ A_2$	$A_1\ A_1\ A_1\ A_1$	
6 -	INC	0 1 1 0	R R R R	Increment contents of register RRRR.
7 -	*ISZ	0 1 1 1	R R R R	Increment contents of register RRRR. Go to ROM address $A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1\ A_1$ (within the same ROM that contains this ISZ instruction) if result = 0, otherwise go to the next instruction in sequence.
- -		$A_2\ A_2\ A_2\ A_2$	$A_1\ A_1\ A_1\ A_1$	
8 -	ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with carry.
9 -	SUB	1 0 0 1	R R R R	Subtract contents of register RRRR to accumulator with borrow.
A -	LD	1 0 1 0	R R R R	Load contents of register RRRR to accumulator.
B -	XCH	1 0 1 1	R R R R	Exchange contents of index register RRRR and accumulator.
C -	BBL	1 1 0 0	D D D D	Branch back (down 1 level in stack) and load data DDDD to accumulator.
D -	LDM	1 1 0 1	D D D D	Load data DDDD to accumulator.
F0	CLB	1 1 1 1	0 0 0 0	Clear both. (Accumulator and carry)
F1	CLC	1 1 1 1	0 0 0 1	Clear carry.
F2	IAC	1 1 1 1	0 0 1 0	Increment accumulator.
F3	CMC	1 1 1 1	0 0 1 1	Complement carry.
F4	CMA	1 1 1 1	0 1 0 0	Complement accumulator.
F5	RAL	1 1 1 1	0 1 0 1	Rotate left. (Accumulator and carry)
F6	RAR	1 1 1 1	0 1 1 0	Rotate right. (Accumulator and carry)
F7	TCC	1 1 1 1	0 1 1 1	Transmit carry to accumulator and clear carry.
F8	DAC	1 1 1 1	1 0 0 0	Decrement accumulator.
F9	TCS	1 1 1 1	1 0 0 1	Transfer carry subtract and clear carry.
FA	STC	1 1 1 1	1 0 1 0	Set carry.
FB	DAA	1 1 1 1	1 0 1 1	Decimal adjust accumulator.
FC	KBP	1 1 1 1	1 1 0 0	Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.
FD	DCL	1 1 1 1	1 1 0 1	Designate command line.

4001/4002/4008/4009/4289 INPUT/OUTPUT AND RAM INSTRUCTIONS

Hex Code	MNEMONIC	OPR D_3 D_2 D_1 D_0	OPA D_3 D_2 D_1 D_0	DESCRIPTION OF OPERATION
2 -	SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X_2 and X_3 time in the instruction cycle.
E0	WRM	1 1 1 0	0 0 0 0	Write the contents of the accumulator into the previously selected RAM main memory character.
E1	WMP	1 1 1 0	0 0 0 1	Write the contents of the accumulator into the previously selected RAM output port. (Output Lines)
E2	WRR	1 1 1 0	0 0 1 0	Write the contents of the accumulator into the previously selected ROM output port. (I/0 Lines)
E3	WPM	1 1 1 0	0 0 1 1	Write the contents of the accumulator into the previously selected half byte of read/write program memory (used with 4008/4009 or 4289 only)
E4	WR0	1 1 1 0	0 1 0 0	Write the contents of the accumulator into the previously selected RAM status character 0.
E5	WR1	1 1 1 0	0 1 0 1	Write the contents of the accumulator into the previously selected RAM status character 1.
E6	WR2	1 1 1 0	0 1 1 0	Write the contents of the accumulator into the previously selected RAM status character 2.
E7	WR3	1 1 1 0	0 1 1 1	Write the contents of the accumulator into the previously selected RAM status character 3.
E8	SBM	1 1 1 0	1 0 0 0	Subtract the previously selected RAM main memory character from accumulator with borrow.
E9	RDM	1 1 1 0	1 0 0 1	Read the previously selected RAM main memory character into the accumulator.
EA	RDR	1 1 1 0	1 0 1 0	Read the contents of the previously selected ROM input port into the accumulator. (I/0 Lines)
EB	ADM	1 1 1 0	1 0 1 1	Add the previously selected RAM main memory character to accumulator with carry.
EC	RD0	1 1 1 0	1 1 0 0	Read the previously selected RAM status character 0 into accumulator.
ED	RD1	1 1 1 0	1 1 0 1	Read the previously selected RAM status character 1 into accumulator.
EE	RD2	1 1 1 0	1 1 1 0	Read the previously selected RAM status character 2 into accumulator.
EF	RD3	1 1 1 0	1 1 1 1	Read the previously selected RAM status character 3 into accumulator.

4004 Instruction Codes

Hex	Mnemonic		Hex	Mnemonic		Hex	Mnemonic		Hex	Mnemonic	
00	–		40	JUN		80	ADD	0	C0	BBL	0
01	–		41	JUN		81	ADD	1	C1	BBL	1
02	–		42	JUN		82	ADD	2	C2	BBL	2
03	–		43	JUN		83	ADD	3	C3	BBL	3
04	–		44	JUN		84	ADD	4	C4	BBL	4
05	–		45	JUN		85	ADD	5	C5	BBL	5
06	–		46	JUN		86	ADD	6	C6	BBL	6
07	–		47	JUN		87	ADD	7	C7	BBL	7
08	–		48	JUN		88	ADD	8	C8	BBL	8
09	–		49	JUN		89	ADD	9	C9	BBL	9
0A	–		4A	JUN		8A	ADD	10	CA	BBL	10
0B	–		4B	JUN		8B	ADD	11	CB	BBL	11
0C	–		4C	JUN		8C	ADD	12	CC	BBL	12
0D	–		4D	JUN		8D	ADD	13	CD	BBL	13
0E	–		4E	JUN	Second hex	8E	ADD	14	CE	BBL	14
0F	–		4F	JUN	digit is part	8F	ADD	15	CF	BBL	15
10	JCN	CN=0	50	JMS	of jump	90	SUB	0	D0	LDM	0
11	JCN	CN=1 also JNT	51	JMS	address.	91	SUB	1	D1	LDM	1
12	JCN	CN=2 also JC	52	JMS		92	SUB	2	D2	LDM	2
13	JCN	CN=3	53	JMS		93	SUB	3	D3	LDM	3
14	JCN	CN=4 also JZ	54	JMS		94	SUB	4	D4	LDM	4
15	JCN	CN=5	55	JMS		95	SUB	5	D5	LDM	5
16	JCN	CN=6	56	JMS		96	SUB	6	D6	LDM	6
17	JCN	CN=7	57	JMS		97	SUB	7	D7	LDM	7
18	JCN	CN=8	58	JMS		98	SUB	8	D8	LDM	8
19	JCN	CN=9 also JT	59	JMS		99	SUB	9	D9	LDM	9
1A	JCN	CN=10 also JNC	5A	JMS		9A	SUB	10	DA	LDM	10
1B	JCN	CN=11	5B	JMS		9B	SUB	11	DB	LDM	11
1C	JCN	CN=12 also JNZ	5C	JMS		9C	SUB	12	DC	LDM	12
1D	JCN	CN=13	5D	JMS		9D	SUB	13	DD	LDM	13
1E	JCN	CN=14	5E	JMS		9E	SUB	14	DE	LDM	14
1F	JCN	CN=15	5F	JMS		9F	SUB	15	DF	LDM	15
20	FIM	0	60	INC	0	A0	LD	0	E0	WRM	
21	SRC	0	61	INC	1	A1	LD	1	E1	WMP	
22	FIM	2	62	INC	2	A2	LD	2	E2	WRR	
23	SRC	2	63	INC	3	A3	LD	3	E3	WPM	
24	FIM	4	64	INC	4	A4	LD	4	E4	WR0	
25	SRC	4	65	INC	5	A5	LD	5	E5	WR1	
26	FIM	6	66	INC	6	A6	LD	6	E6	WR2	
27	SRC	6	67	INC	7	A7	LD	7	E7	WR3	
28	FIM	8	68	INC	8	A8	LD	8	E8	SBM	
29	SRC	8	69	INC	9	A9	LD	9	E9	RDM	
2A	FIM	10	6A	INC	10	AA	LD	10	EA	RDR	
2B	SRC	10	6B	INC	11	AB	LD	11	EB	ADM	
2C	FIM	12	6C	INC	12	AC	LD	12	EC	RD0	
2D	SRC	12	6D	INC	13	AD	LD	13	ED	RD1	
2E	FIM	14	6E	INC	14	AE	LD	14	EE	RD2	
2F	SRC	14	6F	INC	15	AF	LD	15	EF	RD3	
30	FIN	0	70	ISZ	0	B0	XCH	0	F0	CLB	
31	JIN	0	71	ISZ	1	B1	XCH	1	F1	CLC	
32	FIN	2	72	ISZ	2	B2	XCH	2	F2	IAC	
33	JIN	2	73	ISZ	3	B3	XCH	3	F3	CMC	
34	FIN	4	74	ISZ	4	B4	XCH	4	F4	CMA	
35	JIN	4	75	ISZ	5	B5	XCH	5	F5	RAL	
36	FIN	6	76	ISZ	6	B6	XCH	6	F6	RAR	
37	JIN	6	77	ISZ	7	B7	XCH	7	F7	TCC	
38	FIN	8	78	ISZ	8	B8	XCH	8	F8	DAC	
39	JIN	8	79	ISZ	9	B9	XCH	9	F9	TCS	
3A	FIN	10	7A	ISZ	10	BA	XCH	10	FA	STC	
3B	JIN	10	7B	ISZ	11	BB	XCH	11	FB	DAA	
3C	FIN	12	7C	ISZ	12	BC	XCH	12	FC	KBP	
3D	JIN	12	7D	ISZ	13	BD	XCH	13	FD	DCL	
3E	FIN	14	7E	ISZ	14	BE	XCH	14	FE	–	
3F	JIN	14	7F	ISZ	15	BF	XCH	15	FF	–	

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

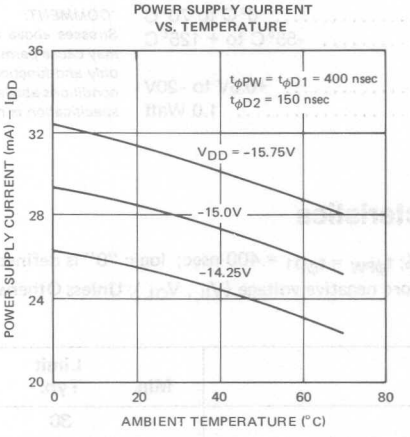
*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ± 5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec; logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless Otherwise Specified.

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Test Conditions
SUPPLY CURRENT						
I_{DD}	Average Supply Current		30	40	mA	T_A=25°C
INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	V_{IL}=V_{DD}
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{ILO}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	4004 TEST Input
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS						
I_{LO}	Data Bus Output Leakage Current			10	μA	V_{OUT}=-12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitance Load
I_{OL}	Data Lines Sinking Current	8	15		mA	V_{OUT}=V_{SS}
I_{OL}	CM-ROM Sinking Current	6.5	12		mA	V_{OUT}=V_{SS}
I_{OL}	CM-RAM Sinking Current	2.5	6		mA	V_{OUT}=V_{SS}
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL}=0.5mA
R_{OH}	Output Resistance, Data Line "0" Level		150	250	Ω	V_{OUT}=V_{SS}-.5V
R_{OH}	CM-ROM Output Resistance, Data Line "0" Level		320	600	Ω	V_{OUT}=V_{SS}-.5V
R_{OH}	CM-RAM Output Resistance, Data Line "0" Level		1.1	1.8	kΩ	V_{OUT}=V_{SS}-.5V
CAPACITANCE						
C_{ϕ}	Clock Capacitance		14	20	pF	V_{IN}=V_{SS}
C_{DB}	Data Bus Capacitance		7	10	pF	V_{IN}=V_{SS}
C_{IN}	Input Capacitance			10	pF	V_{IN}=V_{SS}
C_{OUT}	Output Capacitance			10	pF	V_{IN}=V_{SS}

Typical D.C. Characteristics



A.C. Characteristics

T_A=0°C to 70°C, V_{SS}–V_{DD} = 15V ±5%

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Test Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t_{\phi R}$	Clock Rise Time			50	ns	
$t_{\phi F}$	Clock Fall Times			50	ns	
$t_{\phi PW}$	Clock Width	380		480	ns	
$t_{\phi D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t_{\phi D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_H[3]	Data Bus Hold Time During M_2-X_1 and and X_2-X_3 Transition.	150			ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines
	Data Lines			700	ns	200pF Data Lines[4]
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT}=20pF

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.

2. T_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.

3. All MCS-40 components which may transmit instruction or data to the 4004 at M_2 and X_2 always enter a float state until the 4004 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.

4. $C_{DATA\ BUS}$ = 200pF if 4008 and 4009 or 4289 is used.

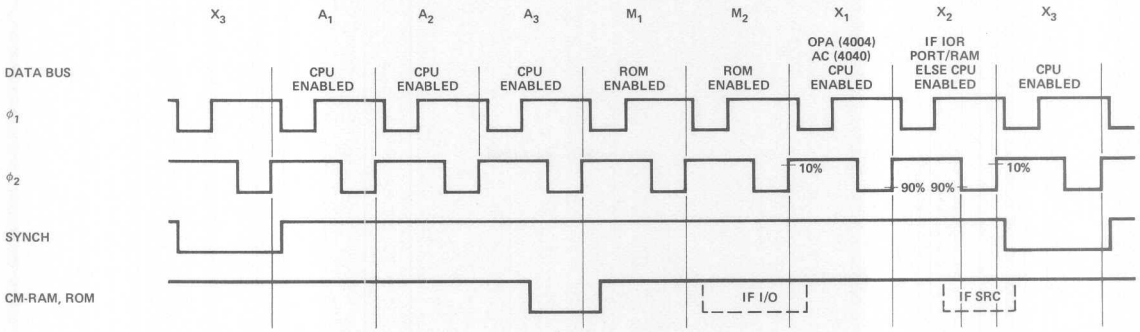


Figure 1. Timing Diagram.

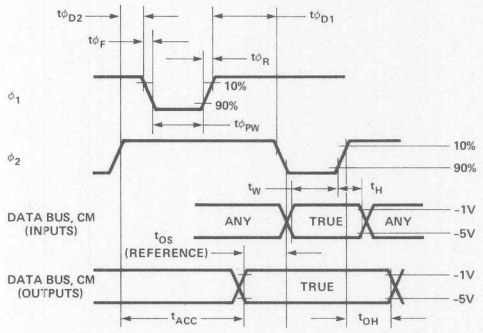


Figure 2. Timing Detail.

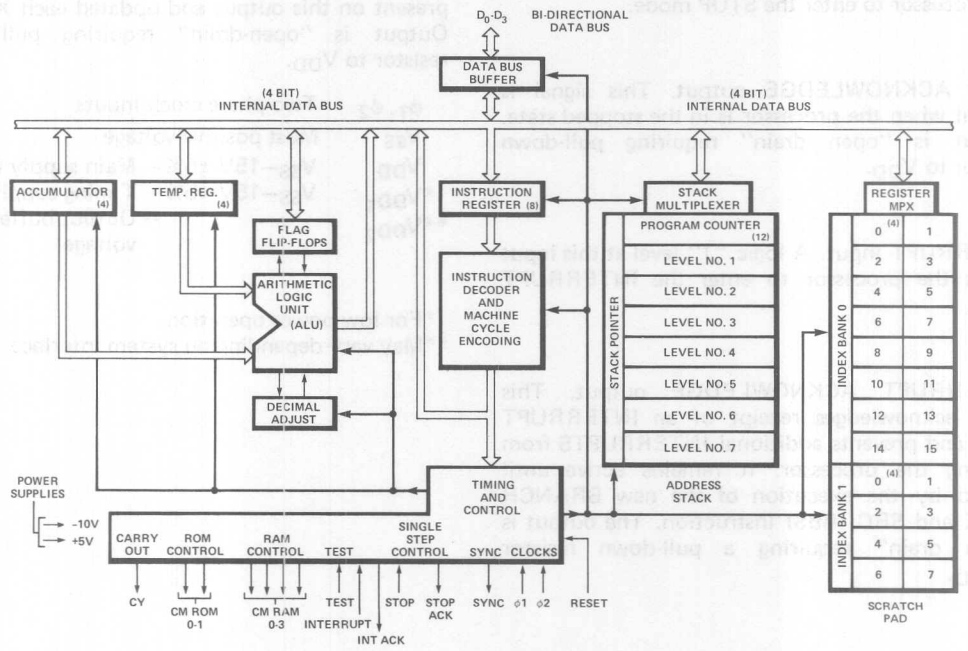
SINGLE CHIP 4-BIT P-CHANNEL MICROPROCESSOR

- **Functionally and Electrically Upward Compatible to 4004 CPU**
- **14 Additional Instructions (60 total) Including Logical Operations and Read Program Memory**
- **Interrupt Capability**
- **Single Step Operation**
- **8K Byte Memory Addressing Capability**
- **24 Index Registers**
- **Subroutine Nesting to 7 Levels**
- **Standard Operating Temperature Range of 0° to 70° C**
- **Also Available With -40° to +85° C Operating Range**

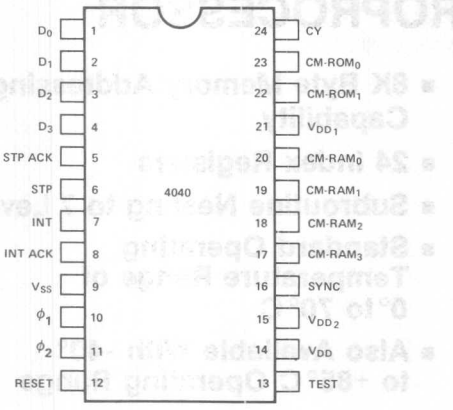
The Intel® 4040 is a complete 4-bit parallel central processing unit (CPU). The CPU can directly address 4K eight bit instruction words or 8K with a bank switch. Seven levels of subroutine nesting, including interrupt, and 24 randomly accessable index registers (24x4) are provided as convenient facilities for the designer. The index registers may be used for addressing or for scratch pad memory for storing computation results. The interrupt feature permits a normal program sequence to be interrupted, with normal program execution continuing after the interrupt service routine is completed. Provisions have also been made to permit single-stepping the CPU using the STOP and ACKNOWLEDGE signals.

The 4040 is an enhanced version of the 4004 and as such retains all the functional capability of that device. It will execute all the 4004 instructions, and is also electrically compatible with all components used with a 4004 CPU.

BLOCK DIAGRAM



Pin Description



D_0-D_3

BIDIRECTIONAL DATA BUS. All address and data communication between the processor and the RAM and ROM chips occurs on these 4 lines.

STP

STOP input. A logic "1" level on this input causes the processor to enter the STOP mode.

STPA

STOP ACKNOWLEDGE output. This signal is present when the processor is in the stopped state. Output is "open drain" requiring pull-down resistor to V_{DD}.

INT

INTERRUPT input. A logic "1" level at this input causes the processor to enter the INTERRUPT mode.

INTA

INTERRUPT ACKNOWLEDGE output. This signal acknowledges receipt of an INTERRUPT signal and prevents additional INTERRUPTS from entering the processor. It remains active until cleared by the execution of the new BRANCH BACK and SRC (BBS) instruction. The output is "open drain" requiring a pull-down resistor to V_{DD}.

RESET

RESET input. A logic "1" level at this input clears all flag and status registers and forces the program counter to zero. To completely clear all address and index registers, RESET must be applied for 96 clock cycles (12 machine cycles).

TEST

TEST input. The logical state of this signal may be tested with the JCN instruction.

SYNC

SYNC output. Synchronization signal generated by the processor and sent to ROM and RAM chips. It indicates the beginning of an instruction cycle.

CM-RAM_0 – CM-RAM_3

CM-RAM outputs. These are bank selection signals for the 4002 RAM chips in the system.

CM-ROM_0 – CM-ROM_1

CM-ROM outputs. These are bank selection signals for program ROM chips in the system.

CY

CARRY output. The state of the carry flip-flop is present on this output and updated each X_1 time. Output is "open-drain" requiring pull down resistor to V_{DD}.

ϕ_1, ϕ_2	Two phase clock inputs
V_{SS}	Most positive voltage
V_{DD}	V_{SS}–15V ±5% – Main supply voltage
*V_{DD_1}	V_{SS}–15V ±5% – Timing supply voltage
**V_{DD_2}	– Output buffer supply voltage

*For low power operation
**May vary depending on system interface

Instruction Set Format

A. Machine Instructions

- 1 word instruction – 8-bits requiring 8 clock periods (1 instruction cycle)
- 2 word instruction – 16-bits requiring 16 clock periods (2 instruction cycles)

Each instruction is divided into two 4-bit fields. The upper 4-bits is the OPR field containing the operation code. The lower 4-bits is the OPA field containing the modifier. For two word instructions, the second word contains address information or data.

The upper 4-bits (OPR) will always be fetched before the lower 4-bits (OPA) during M_1 and M_2 times respectively.

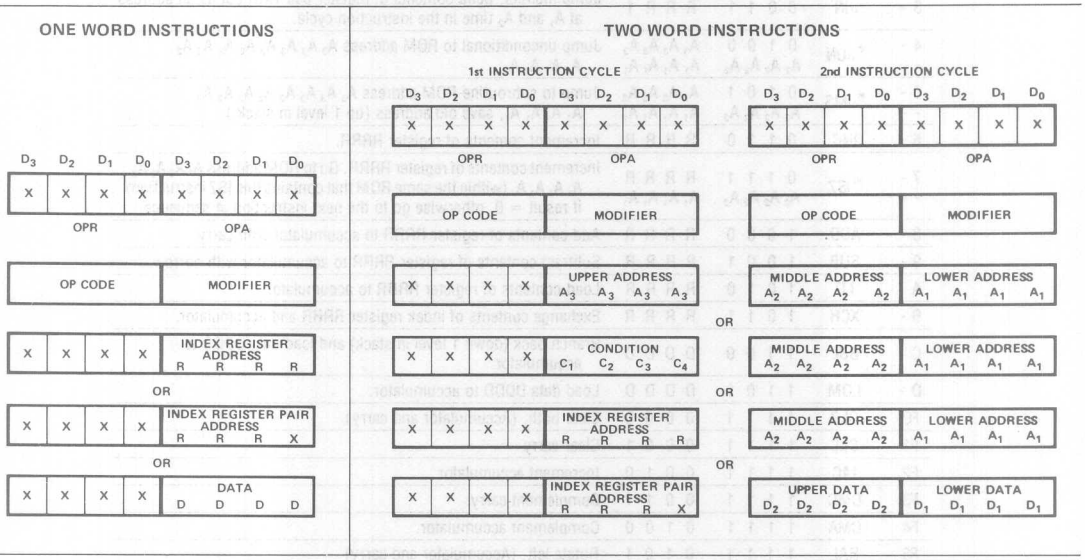


Table I. Machine Instruction Format.

B. Input/Output and RAM Instructions and Accumulator Group Instructions

In these instructions (which are all single word) the OPR contains a 4-bit code which identifies either the I/O instruction or the accumulator group instruction and the OPA contains a 4-bit code which identifies the operation to be performed. Table II illustrates the contents of each 4-bit field.

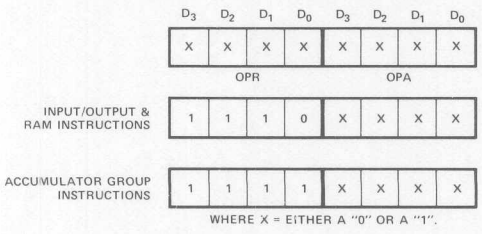


Table II. I/O and Accumulator Group Instruction Formats.

4040 Instruction Set

BASIC INSTRUCTIONS (* = 2 Word Instructions)

Hex Code	MNEMONIC	OPR D_3 D_2 D_1 D_0	OPA D_3 D_2 D_1 D_0	DESCRIPTION OF OPERATION
00	NOP	0 0 0 0	0 0 0 0	No operation.
1 -	*JCN	0 0 0 1	C_1 C_2 C_3 C_4	Jump to ROM address A_2 A_2 A_2 A_2, A_1 A_1 A_1 A_1 (within the same ROM that contains this JCN instruction) if condition C_1 C_2 C_3 C_4 is true, otherwise go to the next instruction in sequence.
- -		A_2 A_2 A_2 A_2	A_1 A_1 A_1 A_1	
2 -	*FIM	0 0 1 0	R R R 0	Fetch immediate (direct) from ROM Data D_2 D_2 D_2 D_2 D_1 D_1 D_1 D_1 to index register pair location RRR.
- -		D_2 D_2 D_2 D_2	D_1 D_1 D_1 D_1	
3 -	FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
3 -	JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A_1 and A_2 time in the instruction cycle.
4 -	*JUN	0 1 0 0	A_3 A_3 A_3 A_3	Jump unconditional to ROM address A_3 A_3 A_3 A_3 A_2 A_2 A_2 A_2 A_1 A_1 A_1 A_1.
- -		A_2 A_2 A_2 A_2	A_1 A_1 A_1 A_1	
5 -	*JMS	0 1 0 1	A_3 A_3 A_3 A_3	Jump to subroutine ROM address A_3 A_3 A_3 A_3 A_2 A_2 A_2 A_2 A_1 A_1 A_1 A_1, save old address (up 1 level in stack.)
- -		A_2 A_2 A_2 A_2	A_1 A_1 A_1 A_1	
6 -	INC	0 1 1 0	R R R R	Increment contents of register RRRR.
7 -	*ISZ	0 1 1 1	R R R R	Increment contents of register RRRR. Go to ROM address A_2 A_2 A_2 A_2 A_1 A_1 A_1 A_1 (within the same ROM that contains this ISZ instruction) if result = 0, otherwise go to the next instruction in sequence.
- -		A_2 A_2 A_2 A_2	A_1 A_1 A_1 A_1	
8 -	ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with carry.
9 -	SUB	1 0 0 1	R R R R	Subtract contents of register RRRR to accumulator with borrow.
A -	LD	1 0 1 0	R R R R	Load contents of register RRRR to accumulator.
B -	XCH	1 0 1 1	R R R R	Exchange contents of index register RRRR and accumulator.
C -	BBL	1 1 0 0	D D D D	Branch back (down 1 level in stack) and load data DDDD to accumulator.
D -	LDM	1 1 0 1	D D D D	Load data DDDD to accumulator.
F0	CLB	1 1 1 1	0 0 0 0	Clear both. (Accumulator and carry)
F1	CLC	1 1 1 1	0 0 0 1	Clear carry.
F2	IAC	1 1 1 1	0 0 1 0	Increment accumulator.
F3	CMC	1 1 1 1	0 0 1 1	Complement carry.
F4	CMA	1 1 1 1	0 1 0 0	Complement accumulator.
F5	RAL	1 1 1 1	0 1 0 1	Rotate left. (Accumulator and carry)
F6	RAR	1 1 1 1	0 1 1 0	Rotate right. (Accumulator and carry)
F7	TCC	1 1 1 1	0 1 1 1	Transmit carry to accumulator and clear carry.
F8	DAC	1 1 1 1	1 0 0 0	Decrement accumulator.
F9	TCS	1 1 1 1	1 0 0 1	Transfer carry subtract and clear carry.
FA	STC	1 1 1 1	1 0 1 0	Set carry.
FB	DAA	1 1 1 1	1 0 1 1	Decimal adjust accumulator.
FC	KBP	1 1 1 1	1 1 0 0	Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.
FD	DCL	1 1 1 1	1 1 0 1	Designate command line.

4040 ONLY INSTRUCTIONS

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
01	HLT	0 0 0 0	0 0 0 1	Executes Halt until interrupt received.
02	BBS	0 0 0 0	0 0 1 0	Return from subroutine and restore SRC.
03	LCR	0 0 0 0	0 0 1 1	Data RAM and ROM bank status loaded into ACC.
04	OR4	0 0 0 0	0 1 0 0	OR accumulator with IR4.
05	OR5	0 0 0 0	0 1 0 1	OR accumulator with IR5.
06	AN6	0 0 0 0	0 1 1 0	AND accumulator with IR6.
07	AN7	0 0 0 0	0 1 1 1	AND accumulator with IR7.
08	DB0	0 0 0 0	1 0 0 0	Select ROM bank 0.
09	DB1	0 0 0 0	1 0 0 1	Select ROM bank 1.
0A	SB0	0 0 0 0	1 0 1 0	Select IR bank 0.
0B	SB1	0 0 0 0	1 0 1 1	Select IR bank 1.
0C	EIN	0 0 0 0	1 1 0 0	Enable interrupt detection.
0D	DIN	0 0 0 0	1 1 0 1	Disable interrupt detection.
0E	RPM	0 0 0 0	1 1 1 0	Load accumulator from 4289-controlled program RAM.

4001/4002/4008/4009/4289 INPUT/OUTPUT AND RAM INSTRUCTIONS

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
2 -	SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X_2 and X_3 time in the instruction cycle.
E0	WRM	1 1 1 0	0 0 0 0	Write the contents of the accumulator into the previously selected RAM main memory character.
E1	WMP	1 1 1 0	0 0 0 1	Write the contents of the accumulator into the previously selected RAM output port. (Output Lines)
E2	WRR	1 1 1 0	0 0 1 0	Write the contents of the accumulator into the previously selected ROM output port. (I/0 Lines)
E3	WPM	1 1 1 0	0 0 1 1	Write the contents of the accumulator into the previously selected half byte of read/write program memory (used with 4008/4009 or 4289 only)
E4	WR0	1 1 1 0	0 1 0 0	Write the contents of the accumulator into the previously selected RAM status character 0.
E5	WR1	1 1 1 0	0 1 0 1	Write the contents of the accumulator into the previously selected RAM status character 1.
E6	WR2	1 1 1 0	0 1 1 0	Write the contents of the accumulator into the previously selected RAM status character 2.
E7	WR3	1 1 1 0	0 1 1 1	Write the contents of the accumulator into the previously selected RAM status character 3.
E8	SBM	1 1 1 0	1 0 0 0	Subtract the previously selected RAM main memory character from accumulator with borrow.
E9	RDM	1 1 1 0	1 0 0 1	Read the previously selected RAM main memory character into the accumulator.
EA	RDR	1 1 1 0	1 0 1 0	Read the contents of the previously selected ROM input port into the accumulator. (I/0 Lines)
EB	ADM	1 1 1 0	1 0 1 1	Add the previously selected RAM main memory character to accumulator with carry.
EC	RD0	1 1 1 0	1 1 0 0	Read the previously selected RAM status character 0 into accumulator.
ED	RD1	1 1 1 0	1 1 0 1	Read the previously selected RAM status character 1 into accumulator.
EE	RD2	1 1 1 0	1 1 1 0	Read the previously selected RAM status character 2 into accumulator.
EF	RD3	1 1 1 0	1 1 1 1	Read the previously selected RAM status character 3 into accumulator.

4040 Instruction Codes

Hex	Mnemonic		Hex	Mnemonic		Hex	Mnemonic		Hex	Mnemonic	
00	NOP		40	JUN		80	ADD	0	C0	BBL	0
01	HLT		41	JUN		81	ADD	1	C1	BBL	1
02	BBS		42	JUN		82	ADD	2	C2	BBL	2
03	LCR		43	JUN		83	ADD	3	C3	BBL	3
04	OR4		44	JUN		84	ADD	4	C4	BBL	4
05	OR5		45	JUN		85	ADD	5	C5	BBL	5
06	AN6		46	JUN		86	ADD	6	C6	BBL	6
07	AN7		47	JUN		87	ADD	7	C7	BBL	7
08	DB0		48	JUN		88	ADD	8	C8	BBL	8
09	DB1		49	JUN		89	ADD	9	C9	BBL	9
0A	SB0		4A	JUN		8A	ADD	10	CA	BBL	10
0B	SB1		4B	JUN		8B	ADD	11	CB	BBL	11
0C	EIN		4C	JUN		8C	ADD	12	CC	BBL	12
0D	DIN		4D	JUN		8D	ADD	13	CD	BBL	13
0E	RPM		4E	JUN	Second hex	8E	ADD	14	CE	BBL	14
0F	—		4F	JUN	digit is part	8F	ADD	15	CF	BBL	15
10	JCN	CN=0	50	JMS	of jump	90	SUB	0	D0	LDM	0
11	JCN	CN=1 also JNT	51	JMS	address.	91	SUB	1	D1	LDM	1
12	JCN	CN=2 also JC	52	JMS		92	SUB	2	D2	LDM	2
13	JCN	CN=3	53	JMS		93	SUB	3	D3	LDM	3
14	JCN	CN=4 also JZ	54	JMS		94	SUB	4	D4	LDM	4
15	JCN	CN=5	55	JMS		95	SUB	5	D5	LDM	5
16	JCN	CN=6	56	JMS		96	SUB	6	D6	LDM	6
17	JCN	CN=7	57	JMS		97	SUB	7	D7	LDM	7
18	JCN	CN=8	58	JMS		98	SUB	8	D8	LDM	8
19	JCN	CN=9 also JT	59	JMS		99	SUB	9	D9	LDM	9
1A	JCN	CN=10 also JNC	5A	JMS		9A	SUB	10	DA	LDM	10
1B	JCN	CN=11	5B	JMS		9B	SUB	11	DB	LDM	11
1C	JCN	CN=12 also JNZ	5C	JMS		9C	SUB	12	DC	LDM	12
1D	JCN	CN=13	5D	JMS		9D	SUB	13	DD	LDM	13
1E	JCN	CN=14	5E	JMS		9E	SUB	14	DE	LDM	14
1F	JCN	CN=15	5F	JMS		9F	SUB	15	DF	LDM	15
20	FIM	0	60	INC	0	A0	LD	0	E0	WRM	
21	SRC	0	61	INC	1	A1	LD	1	E1	WMP	
22	FIM	2	62	INC	2	A2	LD	2	E2	WRR	
23	SRC	2	63	INC	3	A3	LD	3	E3	WPM	
24	FIM	4	64	INC	4	A4	LD	4	E4	WR0	
25	SRC	4	65	INC	5	A5	LD	5	E5	WR1	
26	FIM	6	66	INC	6	A6	LD	6	E6	WR2	
27	SRC	6	67	INC	7	A7	LD	7	E7	WR3	
28	FIM	8	68	INC	8	A8	LD	8	E8	SBM	
29	SRC	8	69	INC	9	A9	LD	9	E9	RDM	
2A	FIM	10	6A	INC	10	AA	LD	10	EA	RDR	
2B	SRC	10	6B	INC	11	AB	LD	11	EB	ADM	
2C	FIM	12	6C	INC	12	AC	LD	12	EC	RD0	
2D	SRC	12	6D	INC	13	AD	LD	13	ED	RD1	
2E	FIM	14	6E	INC	14	AE	LD	14	EE	RD2	
2F	SRC	14	6F	INC	15	AF	LD	15	EF	RD3	
30	FIN	0	70	ISZ	0	B0	XCH	0	F0	CLB	
31	JIN	0	71	ISZ	1	B1	XCH	1	F1	CLC	
32	FIN	2	72	ISZ	2	B2	XCH	2	F2	IAC	
33	JIN	2	73	ISZ	3	B3	XCH	3	F3	CMC	
34	FIN	4	74	ISZ	4	B4	XCH	4	F4	CMA	
35	JIN	4	75	ISZ	5	B5	XCH	5	F5	RAL	
36	FIN	6	76	ISZ	6	B6	XCH	6	F6	RAR	
37	JIN	6	77	ISZ	7	B7	XCH	7	F7	TCC	
38	FIN	8	78	ISZ	8	B8	XCH	8	F8	DAC	
39	JIN	8	79	ISZ	9	B9	XCH	9	F9	TCS	
3A	FIN	10	7A	ISZ	10	BA	XCH	10	FA	STC	
3B	JIN	10	7B	ISZ	11	BB	XCH	11	FB	DAA	
3C	FIN	12	7C	ISZ	12	BC	XCH	12	FC	KBP	
3D	JIN	12	7D	ISZ	13	BD	XCH	13	FD	DCL	
3E	FIN	14	7E	ISZ	14	BE	XCH	14	FE	—	
3F	JIN	14	7F	ISZ	15	BF	XCH	15	FF	—	

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

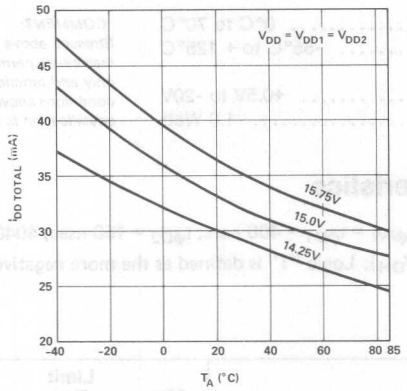
D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec; $t_{\phi D2}$ = 150 nsec; 4040 V_{DD1} = V_{DD2} = V_{DD}; Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}): Unless Otherwise specified.

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
SUPPLY CURRENT						
I_{SB}	Standby Supply Current (V_{DD1} + V_{DD2})		3	5	mA	T_A = 25°C, V_{DD} = V_{SS}
I_{DD} (total)	Supply Current (V_{DD} + V_{DD1} + V_{DD2})		40	60	mA	T_A = 25°C
INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	V_{IL} = V_{DD}
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{ILO}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	4040 TEST and INT inputs
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS						
I_{LO}	Data Bus Output Leakage Current			10	μA	V_{OUT}=-12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitive Load
I_{OL}	Data Lines Sinking Current	8	15		mA	V_{OUT}=V_{SS}
I_{OL}	CM-ROM Sinking Current	6.5	12		mA	V_{OUT}=V_{SS}
I_{OL}	CM-RAM Sinking Current	2.5	6		mA	V_{OUT}=V_{SS}
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL}=0.5mA
R_{OH}	Output Resistance, Data Line "0" Level		150	250	Ω	V_{OUT}=V_{SS}-.5V
R_{OH}	CM-ROM Output Resistance, Data Line "0" Level		320	600	Ω	V_{OUT}=V_{SS}-.5V
R_{OH}	CM-RAM Output Resistance, Data Line "0" Level		1.1	1.8	kΩ	V_{OUT}=V_{SS}-.5V
R_{OH}	INTA, CY, STPA Output Resistance "0" Level		1.1	1.8	kΩ	V_{OUT}=V_{SS}-.5V
CAPACITANCE						
C_{ϕ}	Clock Capacitance		17	25	pF	V_{IN}=V_{SS}
C_{DB}	Data Bus Capacitance		7	10	pF	V_{IN}=V_{SS}
C_{IN}	Input Capacitance			10	pF	V_{IN}=V_{SS}
C_{OUT}	Output Capacitance			10	pF	V_{IN}=V_{SS}

Typical D.C. Characteristics

4040 I_{DD} VS. TEMPERATURE



A.C. Characteristics $T_A = 0°C$ to $70°C$, $V_{SS} - V_{DD} = 15V \pm 5\%$

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t\phi_R$	Clock Rise Time			50	ns	
$t\phi_F$	Clock Fall Times			50	ns	
$t\phi_{PW}$	Clock Width	380		480	ns	
$t\phi_{D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t\phi_{D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_{WRPM}	Data-In Hold Time-RPM Instruction (X_2 state)	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{HRPM}	Data-In Write Time-RPM Instruction (X_2 state)	40	20		ns	
t_H[3]	Data Bus Hold Time During $X_2 - X_3$ Transition (I/O Read Instruction only)	150			ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}[5]	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines
	Data Lines			700	ns	200pF Data Lines[4]
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT}=20pF
t_{DEL}	CY, STPACK, INTACK Delay			2.0	μsec	

NOTES:
1. t_H measured with $t_{\phi R}$ = 10nsec.
2. t_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.
3. All MCS-40 components which may transmit instruction or data to the 4040 at X_2 always enter a float state until the 4040 takes over the data bus at X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.
4. $C_{DATA\ BUS}$ = 200pF if 4008 and 4009 or 4289 is used.
5. The 4040 accumulator is gated out at X_1 time at ϕ_1 leading edge, and the t_{ACC} is 930 nsec + $t_{\phi D2}$.

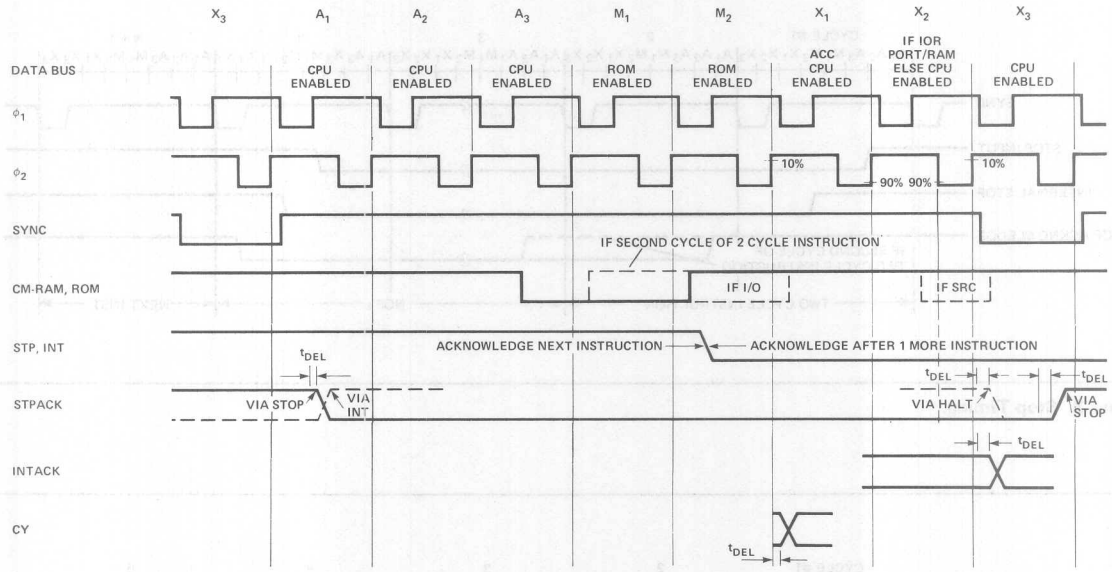


Figure 1. Timing Diagram.

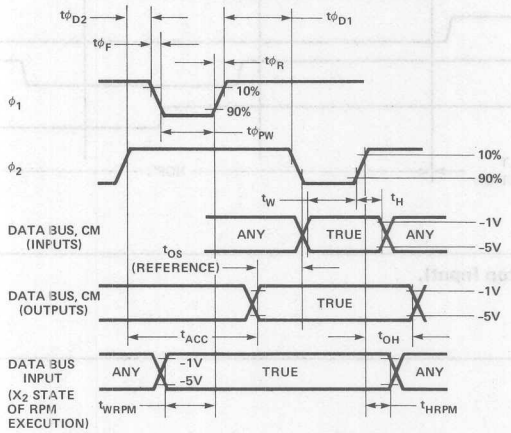


Figure 2. Timing Detail.

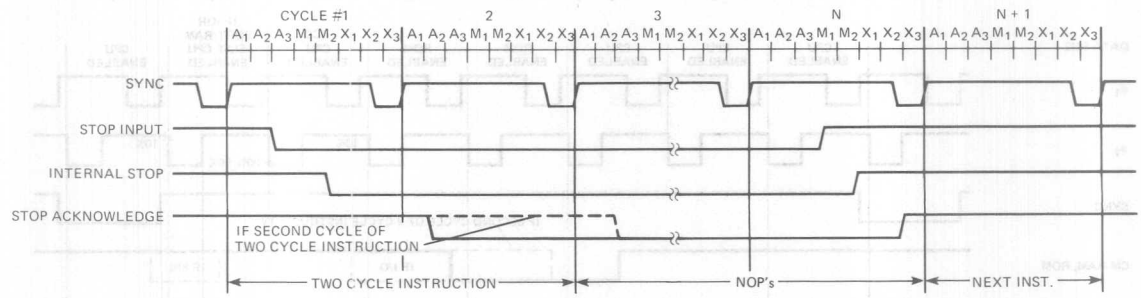


Figure 3. Stop Timing.

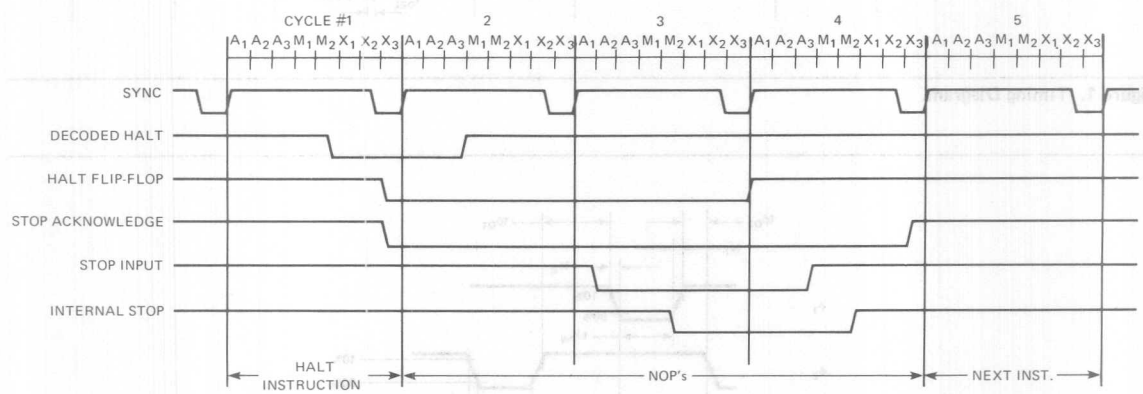


Figure 4. Halt Timing (Exit Using Stop Input).

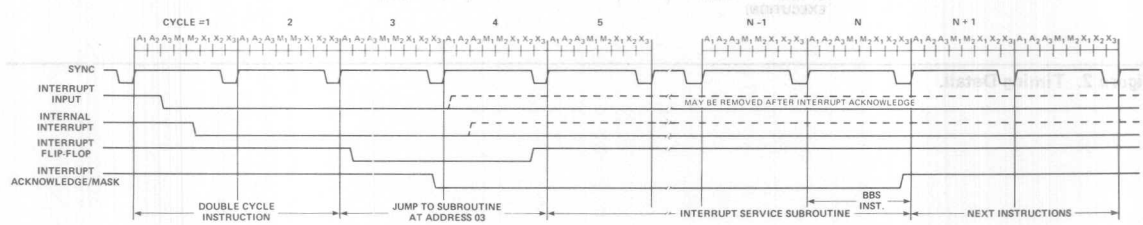


Figure 5. Interrupt Timing.

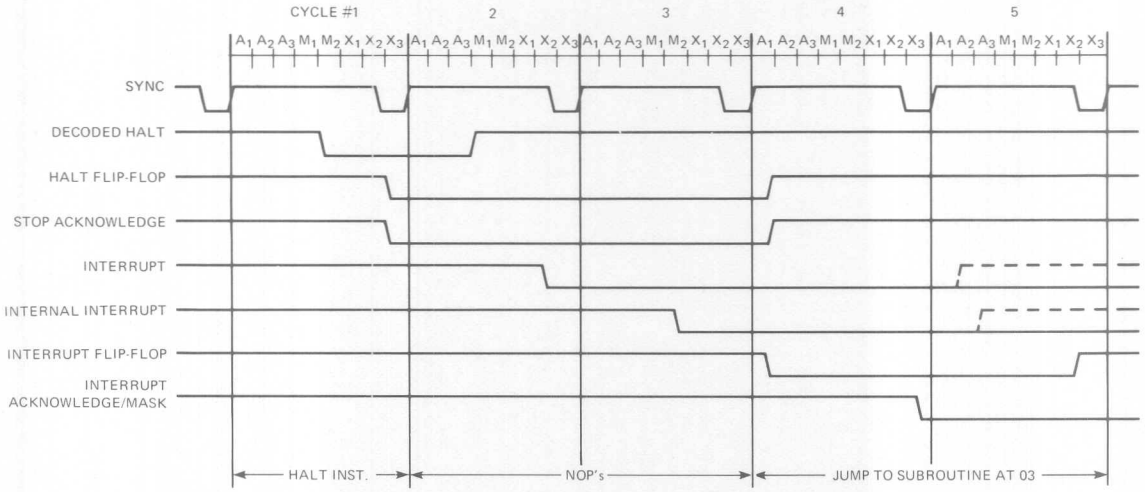
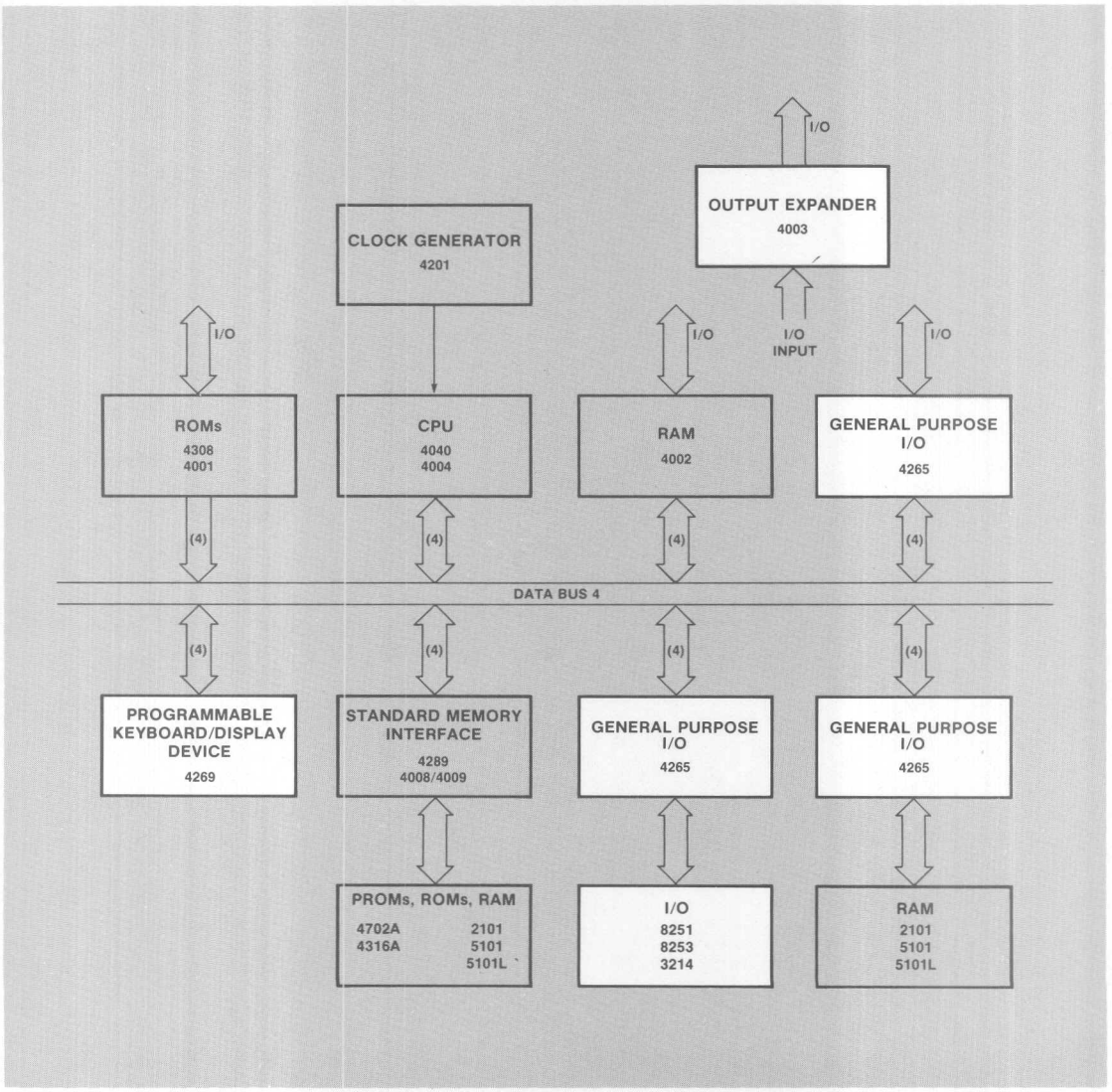


Figure 6. Halt Timing (Exit Using Interrupt).

intel® Microcomputer Systems

I/O Devices

4003 4269
4265 8251
8253



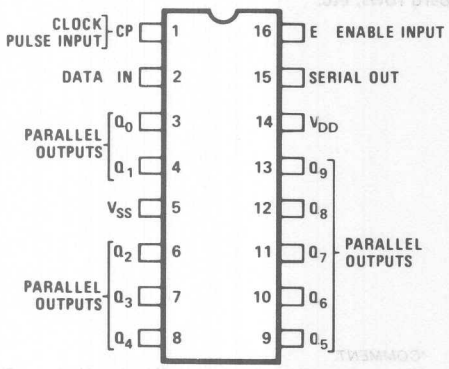
10 BIT SHIFT REGISTER/OUTPUT EXPANDER

- 10 Bit Serial-In/Parallel Out
- Serial-Out Capability for Additional I/O Expansion
- 16 Pin Dual-In-Line Package
- Easy Expansion of I/O Output Capability
- Enable Output Control
- Standard Operating Temperature Range of 0° to 70°C

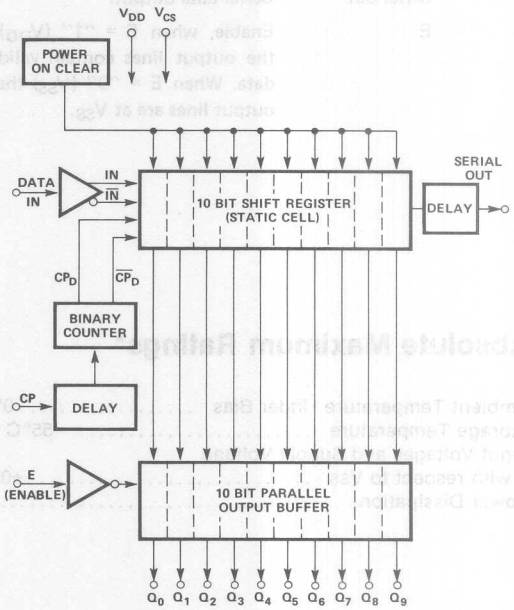
The 4003 is a 10 bit serial-in, parallel-out, serial-out shift register with enable logic. The 4003 is used to expand the number of ROM and RAM I/O ports to communicate with peripheral devices such as keyboards, printers, displays, readers, teletypewriters, etc.

The 4003 is a single phase static shift register; however, the clock pulse (CP) maximum width is limited to 10 msec. Data-in and CP can be simultaneous. To avoid race conditions, CP is internally delayed.

PIN CONFIGURATION



BLOCK DIAGRAM



Pin Description

Pin No.	Designation	Description of Function
1	CP	The clock pulse input. A "0" (V_{SS}) to "1" (V_{DD}) transition will shift data in.
2	DATA IN	Serial data input line.
3	O_0	Parallel data output lines, when enabled. Each pin may be made TTL compatible with a 5.6K pull-down resistor to V_{DD}.
4	O_1	
6	O_2	
7	O_3	
8	O_4	
9	O_5	
10	O_6	
11	O_7	
12	O_8	
13	O_9	
5	V_{SS}	Most positive supply voltage.
14	V_{DD}	Main supply voltage value must be V_{SS} - 15.0V ± 5% (-10v for TTL operation)
15	Serial out	Serial data output.
16	E	Enable, when E = "1" (V_{DD}) the output lines contain valid data. When E = "0" (V_{SS}) the output lines are at V_{SS}.

Functional Description

The 4003 is designed to be typically appended to an MCS-40 I/O port. This can be the I/O port of a 4001, 4002, 4289, 4308, or a 4265. One I/O line is assigned to be the Enable (E), another the Clock (CP), and still another the Serial Data-Input. For example, to access the 4003 a subroutine of sequential outputs consisting of Data, clock pulse on, Enable — followed by an output of clock pulse off and Enable, will serially load the 4003.

Data is loaded serially and is available in parallel on 10 output lines which are accessed through enable logic. When enabled (E = 1 -- V_{DD}), the shift register contents are read out; when not enabled (E = 0 -- V_{SS}), the parallel-out lines are at Logic "0" (V_{SS}). The serial-out line is not affected by the enable logic to allow longer word cascading.

Data is also available serially permitting an indefinite number of similar devices to be cascaded together to provide shift register length multiples of 10.

The data shifting is controlled by the CP signal. An internal power-on-clear circuit will clear the shift register (outputs = 0 or V_{SS}) between the application of the supply voltage and the first CP signal.

The 4003 output buffers are useful for multiple key depression rejection when a 4003 is used in conjunction with a keyboard. In this mode if up to three output lines are connected together, the state of the output is high (Logic "0" or V_{SS}) if at least one line is high.

Another typical application of the 4003 is for Keyboard or Display Scanning where a single bit of Logic "1" is shifted through the 4003 and is used to activate the various digits, keyboard rows, etc.

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. Characteristics

T_A = 0°C to +70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec, $t_{\phi D2}$ = 150 nsec, unless otherwise specified.

Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}), Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}).

SUPPLY CURRENT

Symbol	Parameter	Limit Min.	Typ.[1]	Max.	Unit	Test Conditions
I_{DD}	Average Supply Current		5.0	8.5	mA	t_{WL} = t_{WH} = 8μsec; T_A = 25°C
I/O INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	V_{IL} = V_{DD}
V_{IH}	Input High Voltage	V_{SS}-1.5		V_{SS}+.3		
V_{IL}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	
I/O OUTPUT CHARACTERISTICS						
I_{OL}	Parallel Out Pins Sinking Current, "1" Level	0.6	1.0		mA	V_{OUT} = 0V. For TTL compatibility a 5.6KΩ (±10%) resistor between output and V_{DD} should be added.[2]
I_{OL}	Serial Out Sinking Current, "1" Level	1.0	2.0		mA	V_{OUT} = 0V
V_{OL}	Output Low Voltage	V_{SS}-11	V_{SS}-7.5	V_{SS}-6.5	V	I_{OL} = 10μA
R_{OH}	Parallel-Out Pins Output Resistance "0" Level		400	750	Ω	V_{OUT} = -0.5V
R_{OH}	Serial Out Output Resistance "0" Level		650	1200	Ω	V_{OUT} = -0.5V

Notes: 1. Typical values are to T_A = 25°C and Nominal Supply Voltages.

2. For TTL compatibility on the I/O lines the supply voltages should be V_{DD} = -10V ±5%; V_{SS} = +5V ±5%.

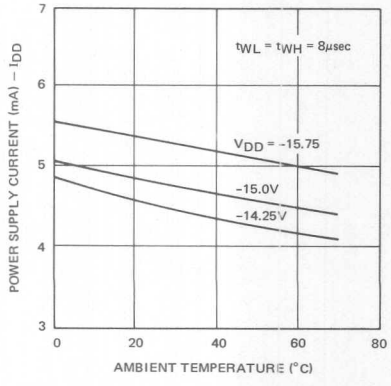
CAPACITANCE

f = 1 MHz; V_{IN} = 0V; T_A = 25°C; Unmeasured Pins Grounded.

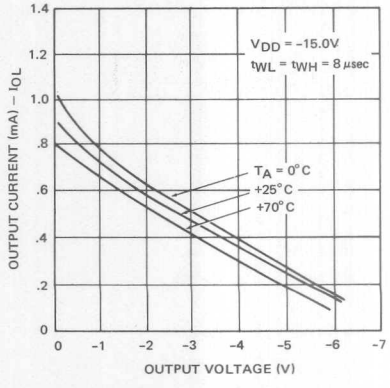
Symbol	Test	Typ.	Max.	Unit
C_{IN}	Input Capacitance	5	10	pF

Typical D.C. Characteristics

POWER SUPPLY CURRENT VS. TEMPERATURE



OUTPUT CURRENT VS. OUTPUT VOLTAGE



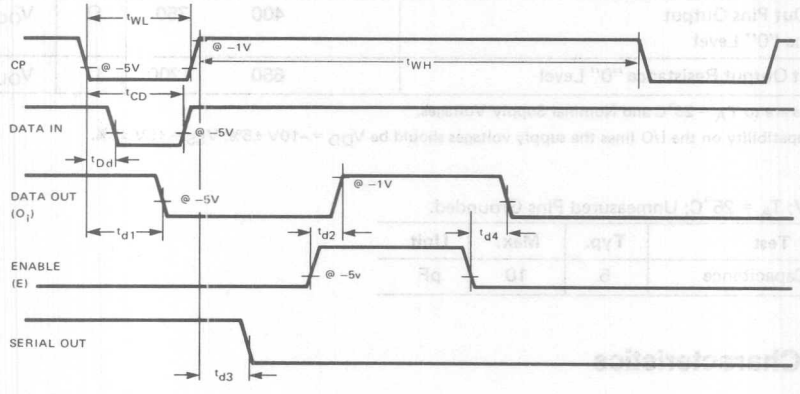
A.C. Characteristics

T_A = 0°C to +70°C; V_{DD} = −15 ±5%, V_{SS} = GND

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
t_{WL}	CP Low Width	6		10,000	μsec	
t_{WH}[1]	CP High Width	6			μsec	
t_{CD}	Clock-On to Data-Off Time	3			μsec	
t_{Dd}[2]	CP to Data Set Delay			250	nsec	
t_{d1}	CP to Data Out Delay	250		1750	nsec	
t_{d2}	Enable to Data Out Delay			350	nsec	C_{OUT} = 20pF
t_{d3}	CP to Serial Out Delay	200		1250	nsec	C_{OUT} = 20pF
t_{d4}	Enable to Data Out Delay			1.0	μsec	C_{OUT} = 20pF

Notes: 1. t_{WH} can be any time greater than 6μsec.
2. Data can occur prior to CP.

Timing Diagram



PROGRAMMABLE GENERAL PURPOSE I/O DEVICE

- Multi-Mode 14 Operating Modes
- 16 Lines of I/O Capability
- Bit Set/Reset
- Multiplexable Outputs
- Eight Bit Transfer Mode
- Interfaces to 8080 Peripherals
- Synchronous and Asynchronous Interface
- Strobed Buffer Inputs and Outputs
- TTL Interface
- Up to Eight 4265s Per System
- Interface to Standard RAMs
- 28 Pin Dual-In-Line Package
- Standard Operating Temperature Range of 0° to 70°C (-40° to +85°C Operating Range to be Available First Quarter 1976)

The 4265 is a general purpose I/O device designed to interface with the MCS-40™ microcomputer family. This device provides four software programmable 4-bit I/O ports which can be configured to allow any one of fourteen unique operating modes for interfacing to data memory or a variety of user peripheral devices.

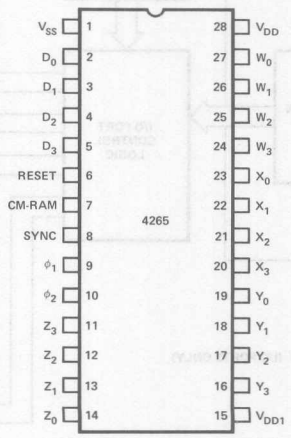
A single MCS-40 system can accomodate up to four 4265s (one per CM-RAM) without external logic or up to eight 4265s with one external decoder.

The 4265 resides on the MCS-40 data bus and uses the same selection procedure as 4002 RAM device. A valid compare selects the 4265 for MCS-40 I/O commands. As in the case of the 4002 or any MCS-40 peripheral circuit, selection occurs only when the proper SRC code and the CM signal are present simultaneously.

The 4265 provides an extremely flexible, general purpose I/O system capable of handling 4- or 8-bit input or output data. One of fourteen basic operating modes can be selected (software programmable) as described below.

Port Z is TTL compatible with any TTL device. Ports W, X, and Y are low-power TTL compatible.

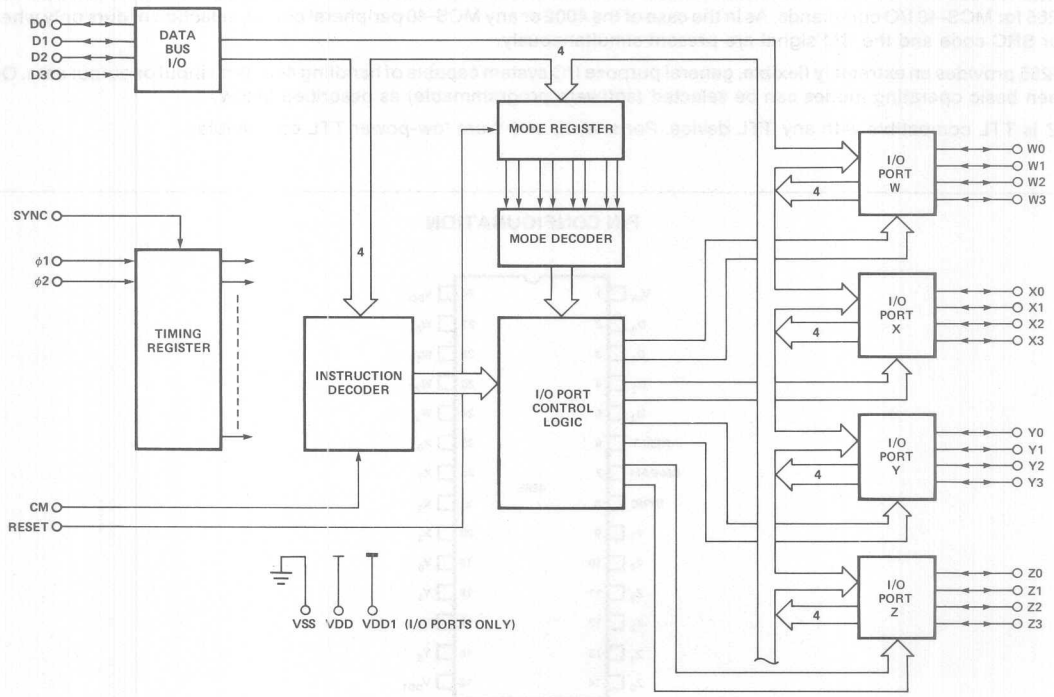
PIN CONFIGURATION



Pin Description

Pin No.	Designation	Function
2-5	D0-D3	Bi-directional data bus. All address, instruction and data communication between processor and I/O ports are transferred on this port.
6	RESET	A negative level (V_{DD}) applied to this pin clears all storage elements, places the 4265 in the Reset Mode and deselects the device.
7	CM	Command input driven by a CM-RAM output of the processor. Used for decoding SRC, RDM, WRM, WMP, SBM, ADM, WR0-3 and RD0-3.
9-10	ϕ1-ϕ2	Non-overlapping clock signals which determine timing.
8	SYNC	Synchronization signal generated by the processor; indicates the beginning of an instruction.
24-27 20-23 16-19 11-14	W3-W0 X3-X0 Y3-Y0 Z3-Z0	Four programmable I/O ports having different functional designation depending on 4265 mode of operation. A data bus "1" negative true (V_{DD}) will appear on a port as a "1" positive true (V_{SS}). These ports are TTL compatible.
28	V_{DD}	Main power supply pin. Value must be V_{SS} -15V ±5%.
15	V_{DD1}	Supply voltage for I/O ports.
1	V_{SS}	Most positive supply voltage (V_{DD1} = 0V, V_{SS} = 5V for TTL I/O ports).
28 = TOTAL PINS		

4265 HARDWARE BLOCK DIAGRAM



4265 PROGRAMMABLE MODES

OPERATING MODES

- Mode 1 – 8-Bit Asynchronous I/O Port (Bidirectional)
  4-Bit Input Port (Unbuffered)
- Mode 2 – 8-Bit Asynchronous I/O Port (Bidirectional)
  4-Bit Output Port
- Mode 3 – 8-Bit Synchronous I/O Port (Bidirectional)
  4-Bit Synchronous Output Port
- Mode 4 – Four 4-Bit Output Ports
- Mode 5 – Three 4-Bit Output Ports
  One 4-Bit Input Port (Unbuffered)
- Mode 6 – Two 4-Bit Output Ports
  Two 4-Bit Input Ports (Unbuffered)
- Mode 7 – One 4-Bit Output Port
  Three 4-Bit Input Ports (Unbuffered)
- Mode 8 – Three 4-Bit Synchronous Output Ports
- Mode 9 – Two 4-Bit Synchronous Output Ports
  One 4-Bit Asynchronous Input Port

OPERATING MODES

- Mode 10 – One 4-Bit Synchronous Output Port
  Two 4-Bit Asynchronous Input Ports
- Mode 11 – Three 4-Bit Asynchronous Input Ports
- Mode 12 – 8-Bit Address Port
  4-Bit Synchronous I/O Port (Bidirectional)
  2 Device Selection Control Signals
- Mode 13 – 8-Bit Address Port
  4-Bit Asynchronous I/O Port (Bidirectional)

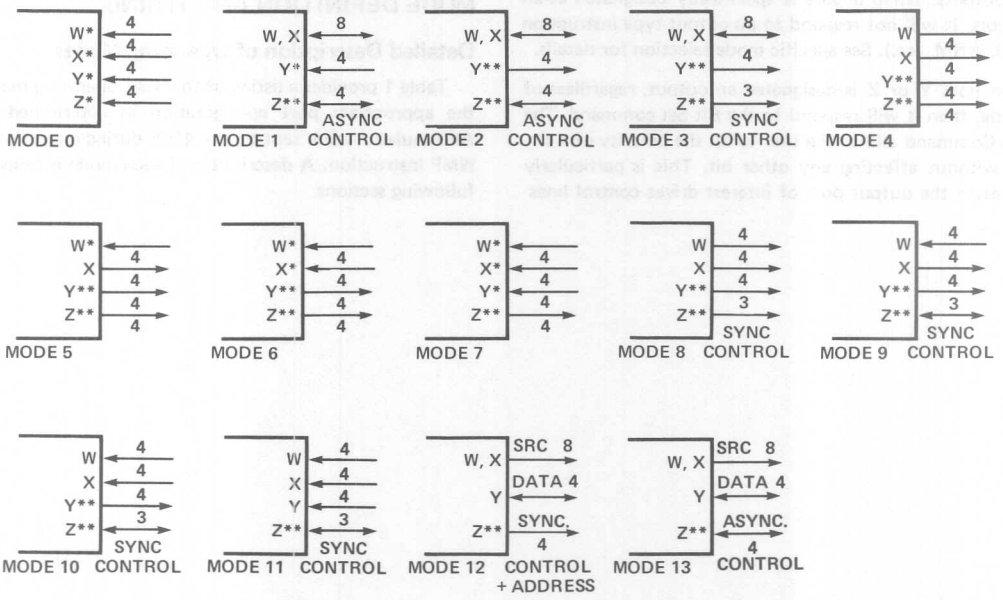
CONTROL AND OPERATING MODE

- Mode 0 – Four 4-Bit Input Ports (Unbuffered)
  Resets I/O Buffers

CONTROL MODES

- Mode 14 – Disables all output buffers, allowing another 4265 to be multiplexed at the port level.
- Mode 15 – Enables output buffers, previous mode restored.

4265 MODE DIAGRAM



* UNBUFFERED INPUT PORTS.

**THE LINES ON THESE PORTS ARE SUBJECT TO THE BIT SET COMMAND.

Functional Description

Control Functions: Two types of operations are possible with the 4265. The device (once selected) can be programmed to one of fourteen basic operating modes. This is accomplished by executing a WMP instruction which sends the 4-bit content of the CPU's Accumulator to the 4265 where it is decoded and used to logically configure the device. A second Control operation makes use of the WRM instruction to select one of eight output lines (Port Y or Z) and perform a SET or RESET operation on that line. This is accomplished by interpreting the 4-bit Accumulator value as follows: The upper three bits select one of eight output latches; the least significant bit determines whether a SET or RESET operation is to be performed.

Data Transfer Functions: The remaining eleven instructions provide four WRITE operations (WR0, WR1, WR2, WR3) and seven READ operations (RD0, RD1, RD2, RD3, ADM, SBM, RDM). These allow data in 4-bit or 8-bit format to be transmitted between the 4265 and external I/O devices or memory devices (all transfers between processor and 4265 are 4-bit transfers).

The sixteen lines of the 4265 are grouped into four ports, four bits each referred to as W, X, Y and Z. The ports can be interrogated by a RD0-3 corresponding to ports W - Z respectively. This means that even when a port is designated as a control port or an output port, the state of the port can be inputted by a RD0-3 instruction (except in modes 12 and 13). The WR0-3 instruction will load the ports W - Z designated outputs. When a port is specifically designated as an input port, it will not respond to an output type instruction (WR0-3, WRM, etc.). See specific mode selection for details.

When port Y or Z is designated an output, regardless of the mode, then it will respond to the Bit Set command. The Bit Set Command allows the user to set the polarity of a single bit without affecting any other bit. This is particularly useful when the output port of interest drives control lines tied to the user system. The user can selectively alter the bit polarity. To alter a bit, the MCS-40 WRM command is utilized.

The 4265 is selected via the CM-RAM line and an appropriate MCS-40™ SRC command. The upper two bits of data at X2 of an SRC instruction with the CM-RAM signal are compared with an address code internal to the 4265. One standard code is available, a code of 2. This allows one 4265 per CM-RAM or up to four per system without additional logic. By using one external decoder and the ability of the DCL (Designate Command Line) instruction to code the CM-RAM lines, up to eight 4265s can be used in a system. Other peripheral devices can share a CM-RAM line with the 4265 (except Mode 12 and 13). For example, a CM-RAM line can contain three 4002 RAMs and one 4265.

The operating modes of the 4265 are selected under program control by the processor. When a 4265 is designed into a specific application, one functional mode is selected. With the possible exception of RESET, ENABLE, and DISABLE, a functional change in mode should not be initiated by the software once the part is designed into a specific application. Since mode selection is done with software, the system's "power up" software routine should sequentially establish the mode of each 4265 prior to "main body" program initiation. The mode selection is accomplished with the accumulator operand of the WMP command.

MODE DEFINITION AND TIMING

Detailed Description of Operating Modes

Table 1 provides a listing of the basic operating modes and the appropriate port configuration as determined by the Accumulator value sent to the 4265 during execution of the WMP instruction. A description of each mode is found in the following sections.

Table 1. Detailed Description of 4265 Operating Modes.

Mode	Port W	Port X	Port Y	Port Z			
0	Input port, unbuffered	Input port, unbuffered	Input port, unbuffered	Input port, unbuffered			
1	Bi-directional; Outputs enabled by signal Z0; When enabled output assumes value loaded by WR0.	Bi-directional; Outputs enabled by signal Z0; When enabled output assumes value loaded by WR1.	Unbuffered input port	Bit 0 Asynchronous input used to enable data out on Ports W, X.	Bit 1 Asynchronous input used to load data to Port W, X input buffers.	Bit 2 Output signal which is normally at V_{SS}. Goes to V_{DD1} on execution of WR 1. Returns to V_{SS} on trailing edge of Z0.	Bit 3 Output signal which is normally at V_{SS}. Goes to V_{DD1} on trailing edge of Z1 and remains at V_{DD1} until execution of RD1.
2	Bi-directional; Output enabled by signal Z0; When enabled output assumes value loaded by WR0.	Bi-directional; Outputs enabled by signal Z0; When enabled output assumes value loaded by WR1.	Buffered output port				
3	Bi-directional; Outputs enabled during WR1 cycle. Output assumes value loaded by WR0.	Bi-directional; Outputs enabled during WR1 cycle. Output assumes value loaded by WR1.	Buffered output port	Synchronous output. Normally at V_{SS}; goes to V_{DD1} during execution of WR 1.	Synchronous output. Normally at V_{SS}; goes to V_{DD1} during RD1 instructions.	Synchronous output. Normally at V_{SS}; goes to V_{DD1} during WR2 instructions.	Unassigned. Line is an output and can be set with WRM. Normally at V_{SS} after mode 3 set.
4	Buffered output port	Buffered output port	Buffered output port	Buffered output port			
5	Unbuffered input port	Buffered output port	Buffered output port	Buffered output port			
6	Unbuffered input port	Unbuffered input port	Buffered output port	Buffered output port			
7	Unbuffered input port	Unbuffered input port	Unbuffered input port	Buffered output port			
8	Buffered output port	Buffered output port	Buffered output port	Output signal normally at V_{SS}; goes to V_{DD1} during WR0.	Output signal normally at V_{SS}; goes to V_{DD1} during WR 1.	Output signal normally at V_{SS}; goes to V_{DD1} during WR2.	Unassigned output. Normally at V_{SS} after mode 8 set.
9	Buffered input port, loaded by signal Z0.	Buffered output port	Buffered output port	Input signal used to load Port W asynchronously.	Output signal normally at V_{SS}; goes to V_{DD1} during WR1.	Output signal normally at V_{SS}; goes to V_{DD1} during WR2.	Unassigned output. Normally at V_{SS} after mode 9 set.
10	Buffered input port, loaded by signal Z0.	Buffered input port, loaded by signal Z1.	Buffered output port	Input signal used to load Port W asynchronously.	Input signal used load Port X asynchronously.	Output signal normally at V_{SS}; goes to V_{DD1} during WR2.	Unassigned output. Normally at V_{SS} after mode 10 set.
11	Buffered input port, loaded by signal Z0.	Buffered input port, loaded by signal Z1.	Buffered input port, loaded by signal Z2.	Input signal used to load Port W asynchronously.	Input signal used to load Port X asynchronously.	Input signal used to load Port Y asynchronously.	Unassigned output. Normally at V_{SS} after mode 11 set.
12	Buffered output port, loaded by SRC instructions—contains upper 4-bits of SRC data.	Buffered output port, loaded by SRC instructions—contains lower 4-bits of SRC data.	Bi-directional; Outputs enabled at any WR instruction; input port unbuffered.	Output signal normally at V_{SS}; goes to V_{DD1} during any WR instruction.	Output signal normally at V_{SS}; goes to V_{DD1} during any RD instruction.	Output signal which is loaded with address bit corresponding to WR or RD operation.	Output signal which is loaded with address bit corresponding to WR or RD operation.
13	Buffered output port, loaded by SRC instructions—contains upper 4-bits of SRC data.	Buffered output port, loaded by SRC instructions—contains lower 4-bits of SRC data.	Bi-directional; Outputs enabled by signal Z0; Inputs loaded by signal Z1.	Asynchronous input used to enable data out on Port Y.	Asynchronous input used to load data to Port Y input buffers.	Output signal normally at V_{SS}; goes to V_{DD1} on execution of WR instruction. Returns to V_{SS} on trailing edge of Z0.	Output signal normally at V_{SS}; goes to V_{DD1} on trailing edge of Z1 and remains at V_{DD1} until execution of RD instruction.
14	All outputs disabled, data saved.	All outputs disabled, data saved.	All outputs disabled, data saved.	All outputs disabled, data saved.	All outputs disabled, data saved.	All outputs disabled, data saved.	All outputs disabled, data saved.
15	Previous information restored.	Previous information restored.	Previous information restored.	Previous information restored.	Previous information restored.	Previous information restored.	Previous information restored.

a. Reset Mode — Mode 0

WMP Operand — 0000

Mode Description: The Reset Mode provides for a programmable reset. Reset will clear all I/O buffers; however, reset will not clear the chip select flip-flop. Hence, the 4265 will remain selected and enabled after a programmable reset. A negative 1 level (V_{DD}) on the RESET pin will cause a response similar to the Reset Mode. The only difference is that the 4265 will be enabled but deselected.

Port Description: Ports W, X, Y, and Z are unbuffered input. Hence, they can be read with RD0-3, transferring the state of the port lines into the accumulator. A positive "1" (V_{SS}) will appear in the accumulator as a negative true "1" (V_{DD}). Port Y will also respond to the RDM, SBM and ADM instructions.

b. 8-Bit Asynchronous I/O Mode with Input — Mode 1

WMP Operand — 0001

Mode Description: The 8-bit I/O mode is used to transfer bi-directional data bytes between the MCS-40™ and the peripheral circuits. Four control lines (Port Z) allow an asynchronous information transfer. Two signals are associated with the input function and two with the output function. Port Y is defined as an unbuffered input.

Port Description

Port W, X — These two ports are combined to transfer 8-bits of I/O under asynchronous control of Port Z. Port W will be loaded with a WR0 and Port X will be loaded with WR1. The WR1 will initiate the write "handshake" on Port Z. When the two ports are interrogated, a sequential RD0 and RD1 will cause the IA line to be deactivated.

Port Y — This port is an unbuffered input, interrogated with an RD2, RDM, ADM or SBM instruction.

Port Z

Z0 OA — Output acknowledge to the 4265 from the users logic. This signal is activated by the users logic (made negative) in response to the OI signal. The OA signal will enable the 4265 output buffer onto Ports W and X. It should be sufficiently long to allow the transfer.

Z2 OI — Output initiate from the 4265. This signal will be generated when Port X has been loaded via a WR1. Port W and Port X should be loaded in the WR0-WR1 sequence. When the OI signal is active, the external device will request data with the OA. The trailing edge of OA will cause the 4265 to remove the OI. If no OA response is received, OI will be active until the next WR0, where it will be removed until the next WR1.

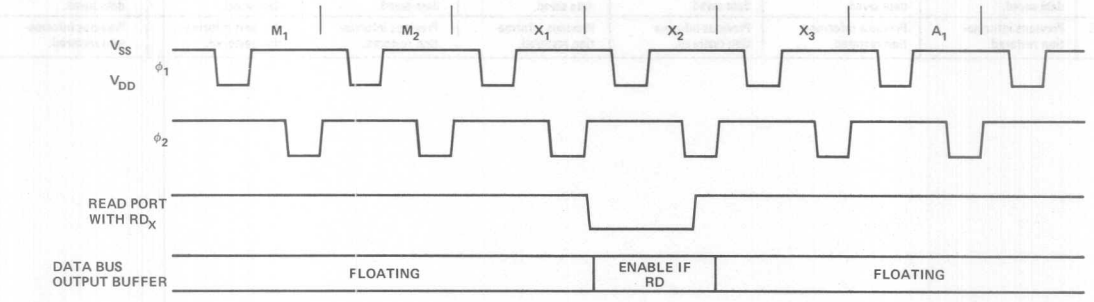


Figure 1. 4265 Mode 0 Timing.

Z1	II	Input initiate to the 4265 from the users logic. The signal will be used as a strobe signal to latch the 8-bit contents of the Port W, X lines into the respective buffers. Data is transferred on the negative to the positive transition. This transition will cause the IA signal to be set.
Z3	IA	Output from the 4265. The IA signal will transition to the positive state when an RD1 command is executed. This indicates that the processor has interrogated Port W, X buffer. The processor should read the data in the sequence of RD0 followed by an RD1.

c. 8-Bit Asynchronous I/O Mode with Output – Mode 2

WMP Operand – 0010

Mode Description: Same as for Mode 1, except Port Y is a buffered output port.

Port Description: Port W, X, Z; same as for Mode 1. Port Y: This port is a buffered output port which can be loaded with a WR2 instruction and can be read by an RD2, RDM, ADM, and SBM.

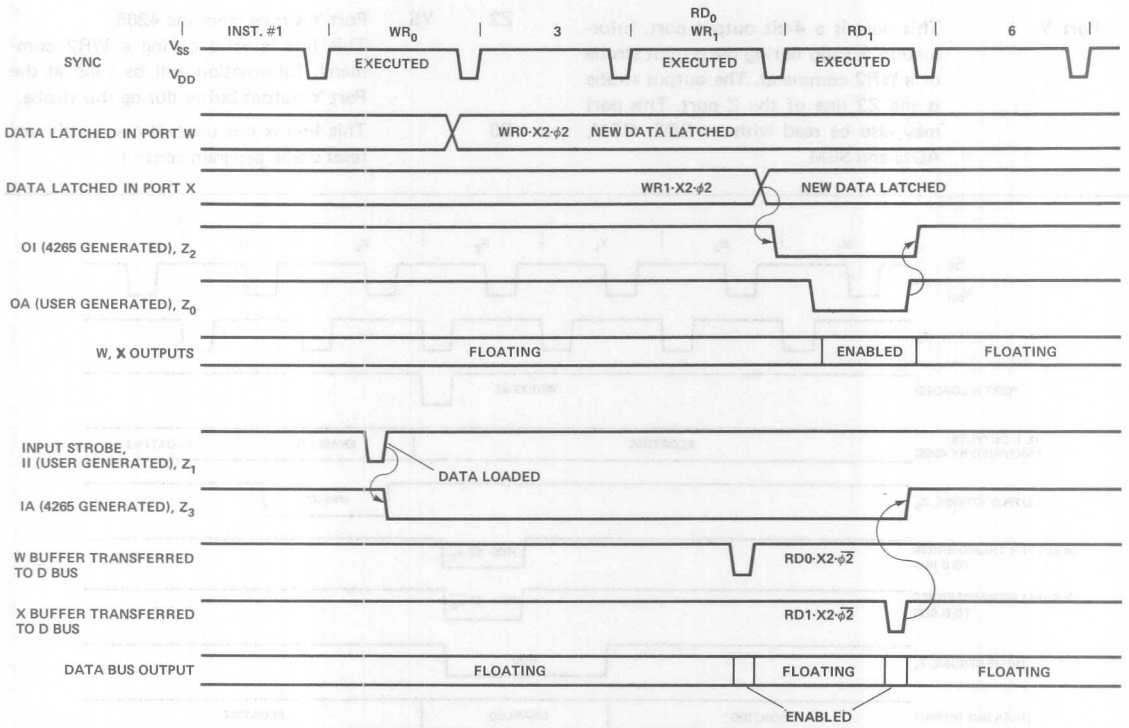


Figure 2. 4265 Modes 1 and 2 Timing.

d. 8-Bit Synchronous I/O Mode with Output – Mode 3

WMP Operand – 0011

Mode Description: This mode is functionally similar to Modes 1 and 2 in terms of its byte transfer feature. However, the transfer control is synchronous. Port W, X are buffered outputs or unbuffered inputs, depending on the direction of transfer. Port Z provides the synchronous strobe control. Port Y is a buffered output port.

Port Description

Port W, X — These two ports are combined to transfer bi-directional 8-bit information under synchronous control. Output data should be loaded into Ports W, X with the WR0-WR1 sequence. The input of information should be sequentially read with an RD0 followed by an RD1.

Port Y — This port is a 4-bit output port. Information is valid during the output strobe of a WR2 command. The output strobe is the Z2 line of the Z port. This port may also be read with an RD2, RDM, ADM and SBM.

Port Z

Z0 OS — Output strobe from 4265. This line is valid during a WR1 command. Information from the output buffers of Ports W and X is present at Ports W and X output lines only during the signal.

Z1 IS — Input strobe from 4265. This line is valid during an RD0 command. Information is taken off the Port W, X lines and is latched in the Port W, X buffers. The RD0 will read the information pertaining to Port W. RD1 will input information pertaining to Port X. The ports must be read by RD0 followed by an RD1. Data will be latched in the W and X Ports with the RD0. Information should be valid at the trailing edge of IS.

Z2 YS — Port Y strobe from the 4265. This line is valid during a WR2 command. Information will be valid at the Port Y output buffer during this strobe.

Z3 — This line is not used. It can be bit set/reset under program control.

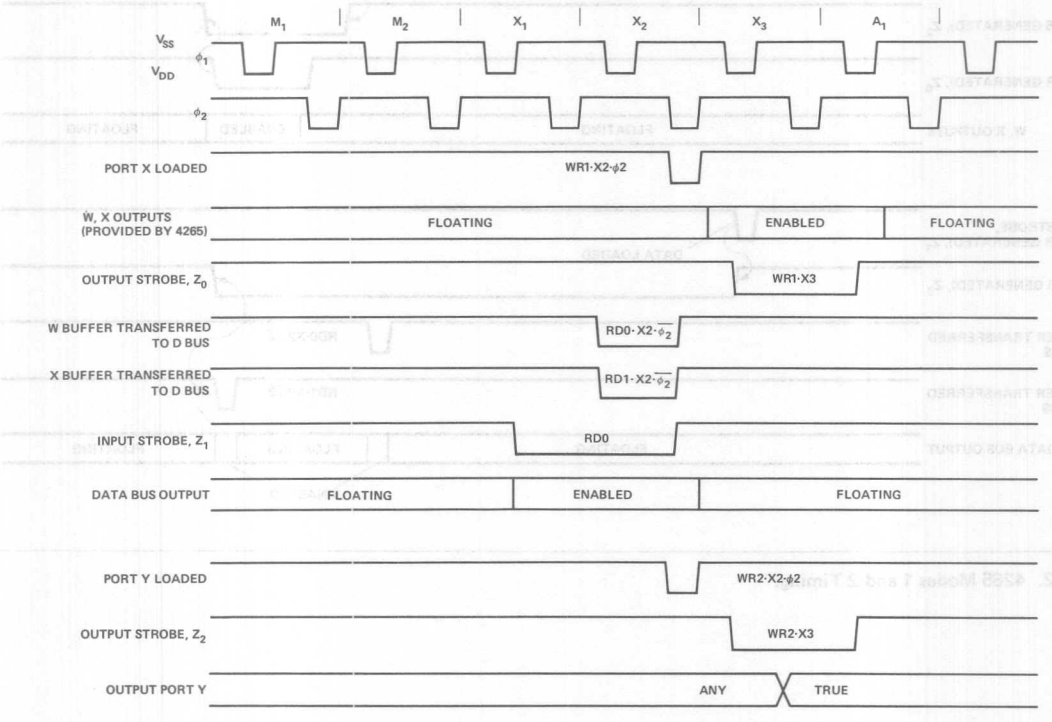


Figure 3. 4265 Mode 3 Timing.

e. Four Port Programmable I/O Modes -- Modes 4-7

WMP Operand -- 0100-0111

Mode Description: These modes consist of four combinations of static buffered outputs and unbuffered inputs. When combined with the Reset Mode, all combinations of inputs and outputs on four ports are possible.

Port Description: The following five modes have static buffered outputs (0) or unbuffered inputs (I).

WMP	Port:	W	X	Y	Z
0100		0	0	0	0
0101		I	0	0	0
0110		I	I	0	0
0111		I	I	I	0
0000 (reset mode)		I	I	I	I

Those ports of Y and Z designated outputs are subject to bit set/reset capability. All output buffers may be read with the respective RDx (RD0-RD3). Port Y will respond to RDM, ADM and SBM in addition to RD2.

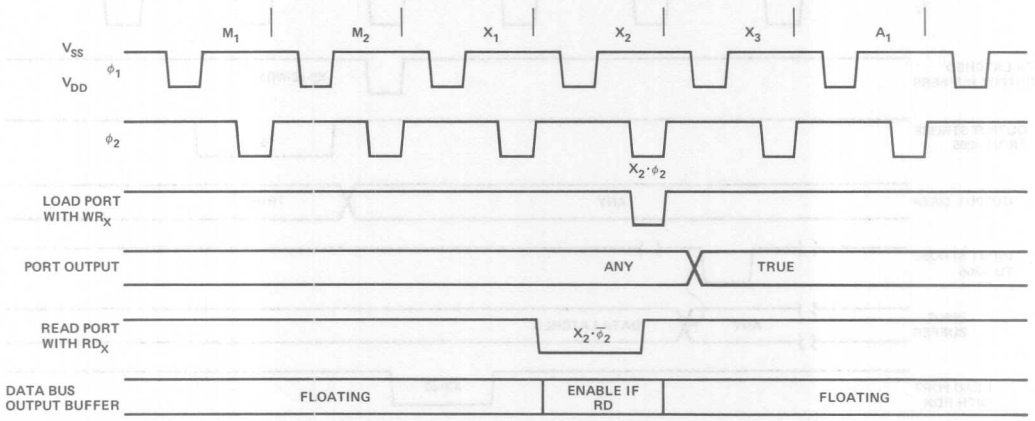


Figure 4. 4265 Modes 4-7 Timing.

f. Three Port Programmable I/O Mode with Synchronous Output and Asynchronous Input Port — Modes 8-11

WMP Operand — 1000-1011

Mode Description: Each 4-bit port can be configured as a buffered input or buffered output port and each has its own control line for synchronizing data transfers. As an example, if in Mode 8, when the processor executes a WR0 instruction, 4-bits of data are transferred to the Port W output buffer and subsequently to the Port W output lines. Output Strobe Z0 serves as a data valid signal which can be used by external logic to latch the data. In Mode 11, Input Strobe Z0 is used to latch the 4-bit data appearing on the Port W lines into the Port W input buffer. The Input Strobe is user generated.

Port Description: The following five modes have synchronous outputs (0) or asynchronous inputs (1):

WMP	Port:	W	X	Y	Z0	Z1	Z2	Z3
1000		0	0	0	W	W	W	X
1001		I	0	0	R	W	W	X
1010		I	I	0	R	R	W	X
1011		I	I	I	R	R	R	X

Where: R = input strobe independent of instruction executed
W = output strobe (WR0-2) from 4265
X = not used

Port Y will respond to RDM, SBM and ADM in the same way as an RD2. Z3 is unused and may be bit set/reset. All output buffers may also be read with the respective RDx.

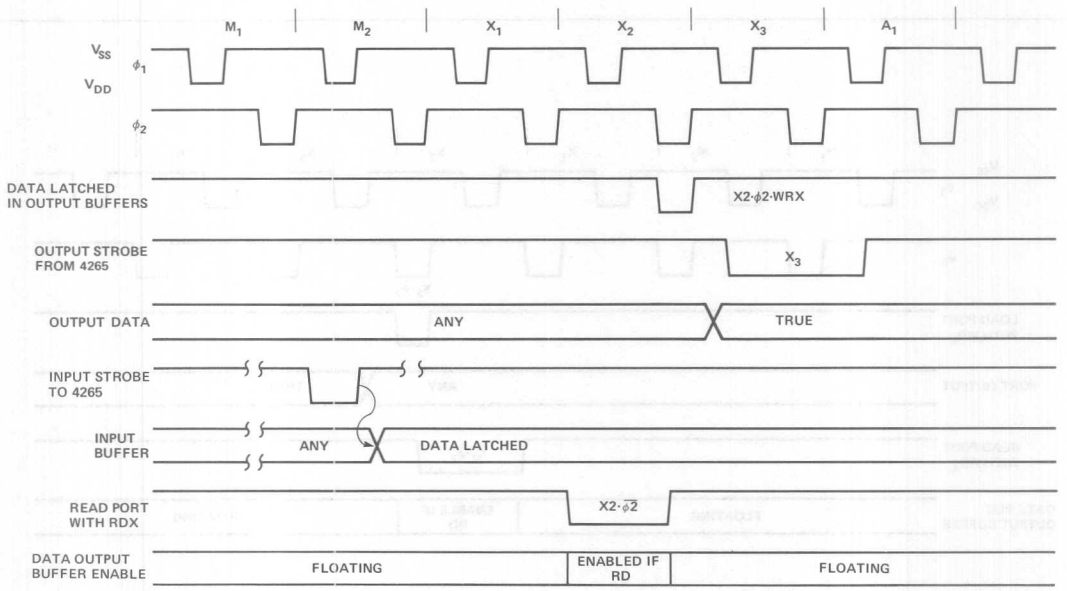


Figure 5. 4265 Modes 8-11 Timing.

g. 4-Bit I/O with 8-Bit SRC Address and 4-Bit Synchronous Control Port – Mode 12

WMP Operand – 1100

Mode Description: In this mode, the most recent 8-bit SRC operand is displayed on Port W and X. The 4265 will treat all SRC instructions as being valid as long as the CM-RAM line for this 4265 has been selected by an appropriate DCL (Designate Command Line) instruction. Ports W and X will change each time they receive an SRC and CM-RAM. The 4-bit data port (Port Y) will perform bi-directional synchronous I/O. The port output buffer may be loaded with a WR0-3 and the port input buffer will be read with RD0-RD3, RDM, SBM or ADM. The control port will provide mutually exclusive input or output strobes depending on the current instruction. Two of the control lines may be used for device selection. This mode can be used to interface up to 1K of external storage (RAM-2111, 4101, 5101) or a multitude of external I/O devices. Once this mode is programmed, all SRC values will not be treated as 4265 selection or deselection instructions.

Port Description

Port W, X		This port will display the most recent SRC and will be altered with each SRC when selected. Otherwise, the output is static.
Port Y		This is a bi-directional data port that will latch data with a RD0-RD3, RDM, ADM, and SBM. The port will output data with a WR0-WR3.
Port Z		
Z0	OS	Output strobe from 4265. Active during WR0-WR3. Data will be valid during this strobe.
Z1	IS	Input strobe from 4265. Active during RD0-RD3, RDM, SBM, and ADM. The leading edge of this strobe will cause the user to provide valid data to be latched by Port Y by the trailing edge of IS.
Z2, Z3		2-bit address port used for memory or device selection. Both lines will be preset to 00 by selection of this mode. They will retain the value of the previous RDx or WRx instruction so that each selection can respond to RDM, SBM and ADM. If, for example an I/O sequence consists of an RD3 followed by an ADM, Z3 and Z2 will be at 11 state by the RD3 and remain in that state for the ADM command. If the third I/O command is a WR0, the Z3 and Z2 will be placed to the 00 state.

Effect of RDx and WRx Instructions:

Z3	Z2	
0	0	RD0, WR0
0	1	RD1, WR1
1	0	RD2, WR2
1	1	RD3, WR3
No Change		RDM, ADM, SBM

(Positive True)

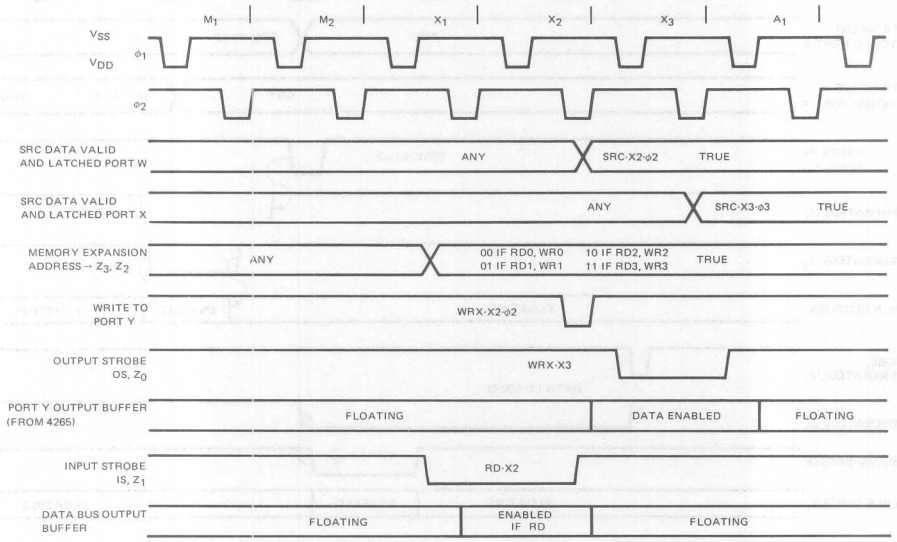


Figure 6. 4265 Mode 12 Timing.

h. 4-Bit I/O Mode with 8-Bit Address Port and 4-Bit Asynchronous Control Port — Mode 13

WMP Operand — 1101

Mode Description: This mode is functionally similar to Mode 12. Port W, X are loaded with the SRC value. Port Y is a bi-directional data port. Port Z is a 4-bit asynchronous control port similar to Mode 1 and 2.

Port Description

Port W, X		Same as Mode 12.
Port Y		Bi-directional port similar to Port W and Port X in mode 1.
Port Z		
Z0	OA*	Output acknowledge to 4265.
Z2	OI*	Output initiate from 4265, active during WRx.
Z1	II*	Input initiate to 4265.
Z3	IA*	Input acknowledge from 4265 active during RDx, RDM, ADM or SBM.

*Refer to Mode 1, Port Z. Note that in mode 13, Port Z controls data transmission in Port Y, not Ports W and X.

i. Disable/Enable

WMP Operands 1110 and 1111 do not cause mode change; they disable or enable the 4265 GP I/O.

WMP 1110 - chip disable:

a. All output buffers are disabled - I/O lines are in floating conditions.
b. The 4265's status (mode, chip select FF, data buffers) is not changed. Hence:
   1. Previous buffered inputs can be read by the CPU from designated ports (a disabled 4265 cannot have its input buffers loaded).
   2. Data on unbuffered inputs can be read directly from external lines.
   3. Previous buffered outputs can be changed on designated ports.
   4. Bit set/reset can be initiated.
   5. Any mode change can be initiated.
   6. The chip can be deselected by an SRC or by a RESET signal.

WMP 1111 - chip enable:
Restoration of normal operation, according to existing mode.

Note: When the 4265 is transferred from reset mode to any other mode, the chip is automatically enabled, so that no programmed enabling is required after reset.

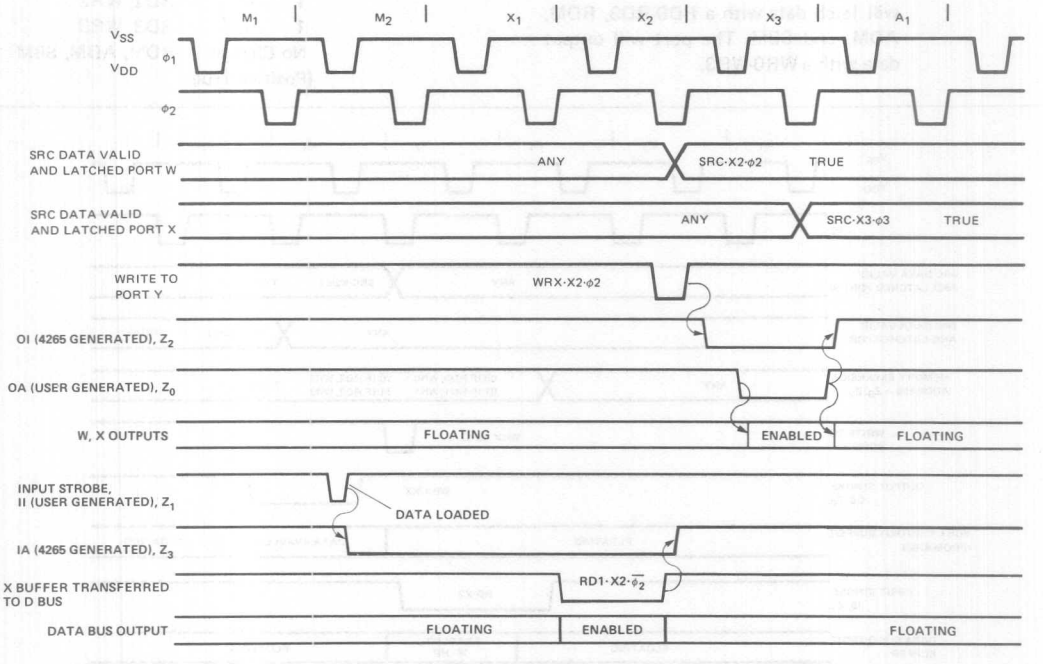


Figure 7. 4265 Mode 13 Timing.

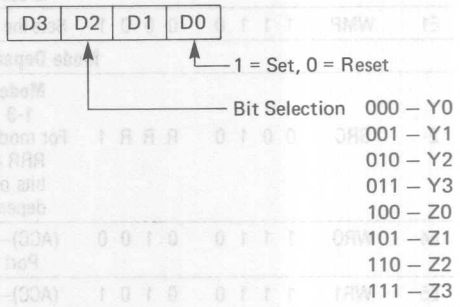
An unselected 4265 can have its input buffers loaded by a user generated strobe if it is in a buffered input mode. A disabled 4265 cannot have its input buffers loaded. Execution of a RDx instruction will result in transfer of the contents of the appropriate input or output buffer for a previously buffered port regardless of whether the 4265 is enabled. If the input was previously unbuffered and the 4265 is disabled, the contents of the port I/O lines will be transferred to the CPU with an RDx. DISABLE and ENABLE do not cause a change from a previously designated mode.

4265 States After Reset and Mode Change

A reset 4265 is automatically enabled and is in Mode 0. If reset occurs by means of external RESET signal, the 4265 will also be deselected. Any mode change which changes Port Z to a control port will reset the Port Z output buffers to their "off" state (V_{SS}). Z_2 and Z_3 in mode 12 are an exception in that these lines go to an inactive state of V_{DD1}. Note that Port Z is a control port in all modes except modes 4-7 and RESET mode. Any mode change which leaves Port Z in a non-control port will leave Port Z output buffers in their previous state.

Bit Set/Reset Operation

This function is performed by decoding the accumulator operand of the WRM instruction. This function can be used in any output port of the programmed configurations and allows individual bit control on Ports Y and Z. Decoding of the WRM operand is as follows:



Care should be taken when bit setting and resetting control bits of Port Z as these bits will also be changing as a function of their synchronous or asynchronous control functions.

4265 I/O Instructions

Table 2 provides a summary of MCS-40 I/O instructions used with the 4265.

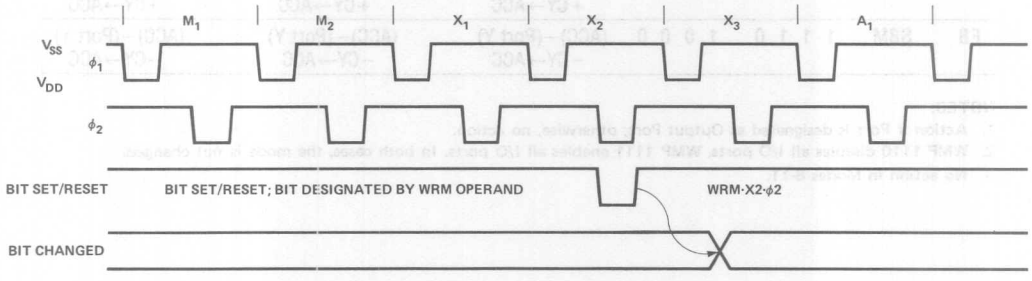


Figure 8. Bit Set/Reset Operation Timing.

Table 2. 4265 I/O Instruction.

Hex Code	MNEMONIC	OPR D_3 D_2 D_1 D_0	OPA D_3 D_2 D_1 D_0	DESCRIPTION OF OPERATION		
Mode Independent Operations						
E0	WRM	1 1 1 0	0 0 0 0	The port Y or port Z bit designated by D_3 D_2 D_1 of the accumulator is set or reset according to D_0 (1=set, 0=reset). [1]		
E1	WMP	1 1 1 0	0 0 0 1	Sets the mode of the 4265 to the value contained in the accumulator. [2]		
Mode Dependent Operations						
				Mode 1-3	**Mode 0, 4-11**	**Mode 12 and 13**
2-	SRC	0 0 1 0	R R R 1	For modes 0-11, the contents of register pair RRR are used to select the 4265 chip (first two bits of first register will contain 10 or 11, depending on chip address)		(RRR_{even})→ Port W (RRR_{odd})→ Port X
E4	WR0	1 1 1 0	0 1 0 0	(ACC)→ Port W	(ACC)→ Port W [1]	(ACC)→ Port Y
E5	WR1	1 1 1 0	0 1 0 1	(ACC)→ Port X	(ACC)→ Port X [1]	(ACC)→ Port Y
E6	WR2	1 1 1 0	0 1 1 0	(ACC)→ Port Y [1]	(ACC)→ Port Y [1]	(ACC)→ Port Y
E7	WR3	1 1 1 0	0 1 1 1	—	(ACC)→ Port Z [1,3]	(ACC)→ Port Y
EC	RD0	1 1 1 0	1 1 0 0	(Port W)→ ACC	(Port W)→ ACC	(Port Y)→ ACC
ED	RD1	1 1 1 0	1 1 0 1	(Port X)→ ACC	(Port X)→ ACC	(Port Y)→ ACC
EE	RD2	1 1 1 0	1 1 1 0	(Port Y)→ ACC	(Port Y)→ ACC	(Port Y)→ ACC
EF	RD3	1 1 1 0	1 1 1 1	(Port Z)→ ACC	(Port Z)→ ACC	(Port Y)→ ACC
E9	RDM	1 1 1 0	1 0 0 1	(Port Y)→ ACC	(Port Y)→ ACC	(Port Y)→ ACC
EB	ADM	1 1 1 0	1 0 1 1	(Port Y)+(ACC) +CY→ACC	(Port Y)+ACC +CY→ACC	(Port Y)+ACC +CY→ACC
E8	SBM	1 1 1 0	1 0 0 0	(ACC)−(Port Y) −CY→ACC	(ACC)−(Port Y) −CY→ACC	(ACC)−(Port Y) −CY→ACC

NOTES:
1. Action if Port is designated as Output Port; otherwise, no action.
2. WMP 1110 disables all I/O ports. WMP 1111 enables all I/O ports. In both cases, the mode is not changed.
3. No action in Modes 8-11.

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400nsec; $t_{\phi D2}$ = 150nsec; V_{DD1} = V_{SS} -5V; Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OH}); Unless Otherwise Specified.

SUPPLY CURRENT

Symbol	Parameter	Limits Min.	Typ.	Max.	Unit	Test Conditions
I_{DD}	Supply Current		35		mA	T_A = 25°C
INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	µA	
V_{IHD}	Data Bus Inputs	V_{SS}-1.5		V_{SS}+.3	V	
V_{IHIO}	I/O Port Inputs	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILD}	Data Bus Inputs	V_{DD}		V_{SS}-5.5	V	
V_{ILIO}	I/O Port Inputs	V_{DD}		V_{SS}-4.2	V	
V_{ILR}	Reset Input	V_{DD}		V_{SS}-4.2	V	
V_{IHR}	Reset Input	V_{SS}-1.5		V_{SS}+.3	V	
OUTPUT CHARACTERISTICS						
V_{OHD}	Data Bus Outputs	V_{SS}-.5	V_{SS}		V	
V_{OHIO}	I/O Port Outputs	V_{SS}-.5			V	I_{OH} = -100µA
V_{OLD}	Data Bus Outputs	V_{SS}-12		V_{SS}-6.5		
V_{OLIO}	I/O Port W,X,Y Outputs			V_{DD1}+.45		I_{OL} = 400µA
V_{OLZ}	I/O Port Z Outputs			V_{DD1}+.45		I_{OL} = 1.6mA

A.C. Characteristics

T_A = 0°C to 70°C, V_{SS} -V_{DD} = 15V ±5%.

Symbol	Parameter	Min.	Typ.	Max.	Unit	Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t\phi_R$	Clock Rise Time			50	ns	
$t\phi_F$	Clock Fall Time			50	ns	
$t\phi_{PW}$	Clock Width	380		480	ns	
$t\phi_{D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t\phi_{D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT} = 20pF

I/O Ports[4]

t_1	Output Settling Time		350		ns	Output Ports
t_{2A}	Output Settling Time		400		ns	Bidirectional I/O Ports (Asynchronous)
t_{2B}	Output Hold Time		400		ns	Bidirectional I/O Ports (Asynchronous)
t_{3A}	Output Settling Time		400		ns	Bidirectional I/O Ports (Synchronous)
t_{3B}	Output Hold Time		100		ns	Bidirectional I/O Ports (Synchronous)
t_{3C}	Output Strobe Write Time		300		ns	Mode 12
t_{3D}	Output Strobe Hold Time		300		ns	Mode 12
t_4	I.S. Delay		200		ns	Z_1, Modes 3, 12
t_5	"Page Select" Outputs Settling Time		600		ns	Z_2, Z_3, Mode 12
t_{6A}	Input Write Time		700		ns	Unbufferred Input Ports(Ports W,X,Y)
t_{6B}	Input Hold Time		0		ns	Unbufferred Input Ports(Ports W,X,Y)

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.
2. T_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.
3. All MCS-40 components which may transmit instruction or data to 4004/4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.
4. For $C_{DATA\ BUS}$ = 500pF, $C_{PORTS\ W,X,Y}$ = 100pF; $C_{PORT\ Z}$ = 50pF.

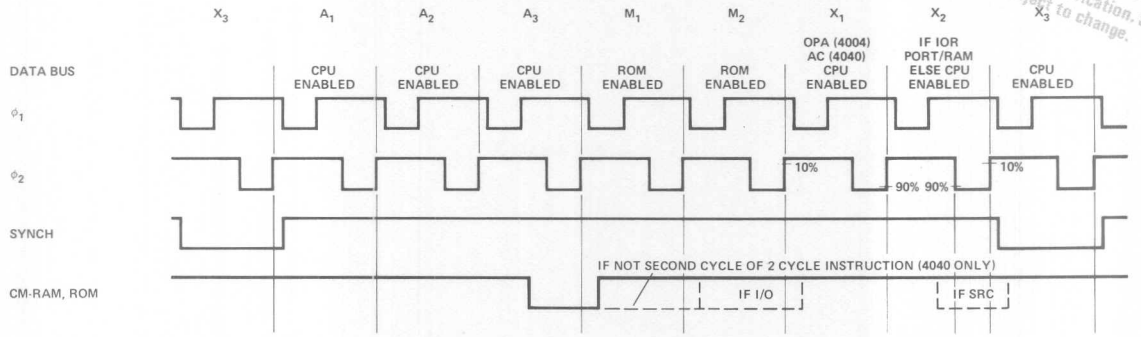


Figure 9. Timing Diagram.

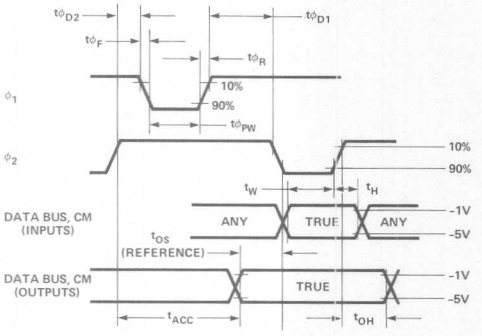


Figure 10. Timing Detail.

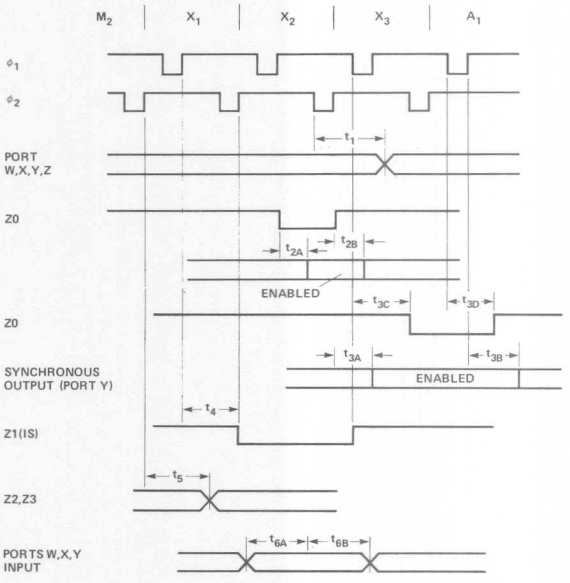


Figure 11. 4265 I/O Timing Diagram.

PROGRAMMABLE KEYBOARD DISPLAY DEVICE

Keyboard Features:

- Programmable to Interface to Encoded Keyboard (8-bit code), 64-Key Scanned Keyboard (expandable to 128 keys) or Sensor Matrix (64 sensors)
- 8 Character FIFO Character Buffer (or RAM in Sensor Mode)
- 2 Key Rollover and Key Debounce
- External Interrupt Line to Indicate When a Character Has Been Entered in Character Buffer

Display Features:

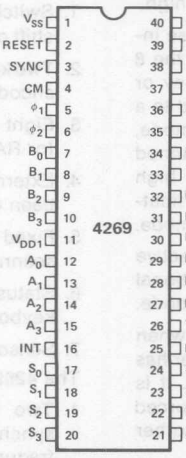
- Programmable to Interface to Individually Scanned Displays or Burrough's Self-Scan* Drive (16, 18, or 20 Characters)
- Two 16 x 4 Display Registers Recirculated Synchronously with Keyboard Scan Lines to Give Automatic Display Refresh
- Display Registers Loadable and Readable Selectively or Sequentially
- 40 Pin Dual In-Line Package
- Standard Operating Temperature Range of 0° to 70°C (-40° to +85°C Operating Range to be Available Second Quarter 1976)

The 4269 has two separate and distinct sections: the keyboard section and the display section. The keyboard section can interface to a range of devices from a matrix of toggle or thumb switches such as found on an instrument panel up to a full typewriter style keyboard. The display section can interface to a range of devices from an array of individual LED indicators up to a gas discharge alphanumeric display.

The 4269 Programmable Keyboard Display (PKD) relieves the 4004 or 4040 CPU from continuously scanning a switch array or refreshing a display under software control. This greatly expands the CPU throughput. The 4269 can scan up to an 8 x 8 keyboard or sensor matrix (or a 2 x 8 x 8 keyboard with the use of the shift or control key input). The display portion can continuously refresh either a single 16 x 8 alphanumeric display; a single 8 x 8 alphanumeric display; a dual 16 x 4 digit display; a single 32 x 4 digit display; a 16 x 6, 18 x 6 or 20 x 6 alphanumeric gas discharge display such as the Burroughs Self-Scan*; or an array of 128 indicators.

*Self-Scan is a registered trademark of the Burrcughs Corporation.

PIN CONFIGURATION



Pin Description

Pin No.	Designation	Function
37-40	D0-D3	Bi-directional data bus. All address, instruction and data communication between the CPU and the PKD are transmitted on these 4 pins.
5-6	ϕ_1-ϕ_2	Non-overlapping clock signals which are used to generate the basic chip timing.
2	RESET	RESET input. A low level (V_{DD}) applied to this input resets the PKD.
1	V_{SS}	Most positive supply voltage.
26	V_{DD}	Main power supply pin. Value must be V_{SS} - 15V ±5%.
3	SYNC	Synchronization input signal driven by SYNC output of the CPU.
4	CM	Command input driven by a CM-RAM output of processor.
17-24	S0-S7	These pins are scan outputs which are used for driving either the key switch or sensor matrix and/or for strobing the display digits. Each line is mutually exclusive, active high (V_{SS}), open drain.
25	RS	The RS pin is toggled for each complete scan of the S drive. This allows for the scan of 16 digits of display data. RS=V_{SS} for the last 8 digits. This line is open drain.
12-15 7-10	A0-A3 B0-B3	These two ports provide two 16 x 4 recirculating display register outputs which are synchronized to the S drive scan. In the gas discharge display mode, A3 is reset and A2 is the clock to the gas discharge display. The 16, 18, or 20 recirculating data characters (6 bits wide) are not synchronized with the S drive scan in the gas discharge mode. These lines are active high.
34-28	R0-R7	These pins are the return sense inputs which are connected to the 8 drive lines via the scanned key or sensor matrix. They are pulled to a low state (V_{DD}) in the sensor mode, pulsed low (V_{DD}) in the scanned keyboard mode, and pulled high upon switch closure. They are floating in the encoded keyboard mode.
35	SHIFT	This is the shift input. It is active high (V_{SS}). This pin is functional only in the scanned keyboard mode.
16	INT	This output is used to indicate when a keyboard or sensor character has been entered into the buffer. It is active low (V_{DD1}), open-sourced and may be "OR" -ed with other 4040 interrupt inputs.
11	V_{DD1}	Supply voltage for display register ports A and B and INT.
36	S/C	This pin is the control key input from the keyboard in the scanned mode. In encoded keyboard mode, this pin can be used to input the strobe pulse from an external keyboard encoder. The strobe should be an active high pulse.

Functional Description

General

The 4269 Programmable Keyboard/Display (PKD) device provides an intelligent interface between an MCS-40 CPU and the keyboard and display portions of an MCS-40 design. The 4269's functions thus allow the use of sophisticated keyboards and displays without placing a large load on the CPU.

The MCS-40 data bus will provide the path for information transfer between the PKD and the 4040 or 4004 CPU. The PKD can be programmed to operate in one of three input modes and one of four output modes as defined by an instruction from the CPU. The modes are:

Input	Sensor, Scanned Keyboard, Scanned Encoded Keyboard
Output	Individually Scanned Display Drive Self-Scan Drive: 16 Characters 18 Characters 20 Characters

The 4269 resides on a CM-RAM line of an MCS-40 system and has a fixed RAM address, #1. Hence, there can be up to four PKD per system without additional logic, one per CM-RAM. The PKD can be accessed with the MCS-40 I/O instruction set to interrogate the keyboard buffer FIFO/sensor RAM and load or read the display registers.

The following is a list of the major keyboard features of the 4269:

1. Switch matrix, organized as an 8 x 8 scanned matrix with shift or control inputs allowing for up to 128 key inputs.
2. Two key roll over; N-key roll over capability if provided by encoded keyboards.
3. Eight character first-in-first-out (FIFO) character buffer (or RAM in the Sensor Mode).
4. External interrupt line to indicate when a character has been entered in the buffer.
5. Fixed key bounce delay of approximately 11 msec in the scanned keyboard mode @ 740 kHz MCS-40 clocks.
6. Status buffer to indicate the number of characters in the keyboard FIFO and keyboard character over-entry.
7. Sensor matrix interface with up to 64 intersections.

The 4269's major display features are:

1. Two 16 x 4 display registers which are recirculated synchronously with keyboard scan lines (at a scan frequency of 180 Hz). This allows for a free standing, scanned readout composed of individual displays.

2. Capability to drive 16, 18, or 20 character gas discharge displays directly via a 20 x 6 display register.
3. Registers are loadable and readable selectively or sequentially.

Mode Selection

The CPU communicates with the 4269 PKD by first selecting it with an SRC (Send Register Control) instruction. The first two bits of the index register pair referenced by the SRC contain 01, the binary address of the 4269 on the CM-RAM line. The 4269 is disabled until it is addressed by a first SRC. After the first SRC, a WR0 instruction is used to set the keyboard and display modes of the 4269 PKD. The CPU's accumulator will contain the information used for setting the PKD modes. The definition of a WR0 as used for a 4269 is given below:

Mnemonic	Instruction Code
WR0	1110 0100

Set the input mode and output mode of the 4269 according to the value contained in the accumulator as follows:

D_3D_2

0 0 Individual, Scanned Displays
0 1 Gas Discharge, 20 Characters
1 0 Gas Discharge, 18 Characters
1 1 Gas Discharge, 16 Characters

D_1D_0

0 0 Sensor, Scanned
0 1 Scanned Keyboard
1 0 Encoded Keyboard, Not Scanned
1 1 Not Used

After the 4269 has been reset by the external RESET signal, the keyboard input mode is set to scanned keyboard mode and the display output mode is set to gas discharge, 16 character mode. Thus, if these modes are the desired input and output modes, it is not necessary to execute the WR0 mode setting instruction.

Internal Display Registers and Pointer

The 4269 has two 16x4 display registers referred to as Display Register A and Display Register B. These two registers can be operated in the individual, scanned display mode as:

1. Two 16 x 4 hexadecimal displays;
2. One 32 x 4 hexadecimal display;
3. One 8 x 8 alphanumeric display;
4. One 16 x 8 alphanumeric display; or
5. An array of 128 indicators.

In the gas discharge modes, the A and B registers are combined and operated as a 6 x 16, 6 x 18 or 6 x 20 register. For a given 6-bit character, the least significant 4-bits will be located in a 4-bit B register location and the two most significant bits in D_1 and D_0 of the corresponding A register location.

For operations on the display registers, the 4269 PKD maintains an internal display register pointer which points to a 4-bit character in the A or B display register.

For the individual, scanned display mode, CPU I/O instructions can be addressed to either Display Register A or Display Register B, according to the register selected by an SRC instruction preceding the I/O instruction. The internal display register pointer can then be set or incremented for addressing characters in either the A or B register.

For gas discharge modes, the internal pointer can be automatically incremented, in an alternating pattern between registers A and B. The alternation pattern is A_0, B_0, A_1, B_1, etc.

In the individual, scanned display mode, the 4-bit characters of Display Register A are outputted on the A_0-A_3 lines. The 4-bit characters of Display Register B are outputted on the B_0-B_3 lines. In the gas discharge modes, the A_0-A_1 and B_0-B_3 lines output the 6-bit character. The A_2 line serves as the clock to the gas discharge display and the A_3 line as the reset to the display.

Synchronization of Scan and Return Lines

In the scanned keyboard and scanned sensor modes a logical one is shifted through a field of zeros in eight Scan(S) lines. Each S Scan line can be used to source a row of eight keys or sensors. All rows of the contact keyboard or sensor matrix will be OR-tied to the eight Return (R) lines. Thus, since only one row will be enabled due to the synchronized ones in the Scan lines, each row of the keyboard or sensor matrix can be read into the Return lines and stored in the Keyboard FIFO/Sensor RAM at the proper RAM location. The 4269 will control all of these operations automatically once it is set to the appropriate keyboard mode.

The Scan Lines are also used in the individual, scanned display mode to select one of eight display characters. The display character itself will be outputted on the A_0-A_3 or B_0-B_3 output lines. The RS output line, which is toggled for each complete scan of the S lines, allows one of sixteen A or B register display characters to be addressed. Again, the 4269 will automatically control the operation of the S and RS lines to continuously read out the characters in the 4269's internal A and B Display Registers and thus continuously refresh the actual display devices.

Note that the Scan lines can be used with both the keyboard and display interfaces since both functions require the same function, i.e., a synchronized shifting of a logical one through a field of zeros.

Software Operation

The WR0 operates on the 4269 PKD completely independent of mode as it actually sets the mode as has already been described. The WR3 is mode independent except for a blanking code and operates as shown below:

WR3

Clears the keyboard/display logic and fills the display RAM with all blanks. The display outputs are also blanked. (Blank code is all logical "1"s for individual, scanned display mode and hex 20 for the gas discharge modes.)

MODE SPECIFIC OPERATIONS

Individual, Scanned Display Mode

The instructions which are used in the individual, scanned display mode are described below:

Mnemonic Instruction Code
SRC 0010 RRR1

The contents of the register pair RRR are used to select the 4269. An SRC is interpreted for individual, scanned display mode as follows:

RRR_{even}	RRR_{odd}	
$D_3D_2D_1D_0$	$D_3D_2D_1D_0$	
0 1 0 0	$n_3n_2n_1n_0$	Selects one of 16 display register characters of Display Register A with the A output lines outputting display characters synchronized with the S Scan lines.
0 1 0 1	$n_3n_2n_1n_0$	Selects one of 16 display register characters of Display Register B with the B output lines outputting display characters synchronized with the S Scan lines.
0 1 1 0	$n_3n_2n_1n_0$	Selects one of 16 display register characters of Register A with Register A output lines being placed at V_{SS} level.
0 1 1 1	$n_3n_2n_1n_0$	Selects one of 16 display register characters of Register B with Register B output lines being placed at V_{SS} level.

WR1 1110 0101

Resets the internal display register pointer to 0 and forces display memory to blank state. Upper two bits of ACC select length of display as follows:

D_3

0 Display B is 16 nibbles deep.
1 Display B is 8 nibbles deep.

D_2

0 Display A is 16 nibbles deep.
1 Display A is 8 nibbles deep.

WRM 1110 0000

Loads the contents of the register addressed by the internal display register pointer with the contents of ACC; then advances the displayed data by one digit in relation to the scan line timing and increments the display register pointer.

RDM 1110 1001

Loads ACC with the contents of the register addressed by the display register pointer and then increments the display register pointer.

WMP 1110 0001

Loads the contents of the register addressed by the display register pointer with the contents of ACC.

RD3 1110 1111

Loads ACC with the contents of the display register pointed to by the display register pointer.

NOTES:

1. If Display A or B is set to 8 nibbles deep, each digit of the display will have double the ON duty-cycle that it would have in the 16 nibble deep setting (360 Hz scan cycle vs. 180 Hz for 16 nibble deep).
2. External resetting initializes the Display A and Display B configurations to 16 nibbles deep.
3. The displayed nibbles in the 8 deep configuration will be from the least significant 8 characters of the display register. The remaining eight words remain available for random data storage by the CPU.
4. The internal display register pointer will increment through all 16 register words, regardless of the display length (8 or 16) for WRM/RDM instructions unless the pointer is reset by an appropriate SRC instruction. In the WRM case, the Display Register A or B's entire contents (used and unused portions) will be rotated.
5. An interface to a 32 x 4 hexadecimal display requires only that software recognize the A and B Display registers as the upper and lower halves of a single display.
6. An interface to a 16 x 8 alphanumeric display requires that software load the upper and lower 4-bits in the A and B registers in an appropriate alternating pattern. SRC instructions will have to proceed each load or read instruction to select the A or B half of the character.
7. If the LSD of a 16 character display is assigned to be the 15th character scanned ($S_7 = V_{SS}$ and $RS = V_{SS}$), and the MSD, the first character (#0) scanned ($S_0 = V_{SS}$ and $RS = V_{DD}$), and if loading is started at display register character 0, successive WRM instructions will shift the display data from the LSD to the MSD as in a calculator. Note that data will then be read back MSD to LSD with the RDM instruction, starting at register 0.

Gas Discharge Modes

The instructions which are used in the gas discharge display modes are described below.

Mnemonic Instruction Code
SRC 0010 RRR1

The contents of the register pair RRR are used to select the 4269. An SRC is interpreted for gas discharge modes as follows:

RRR_{even}	RRR_{odd}	
$D_3D_2D_1D_0$	$D_3D_2D_1D_0$	
0 1 0 0	$n_3n_2n_1n_0$	Selects the nth display register character of Display Register A with display outputs continuing to output the contents of Display Registers A and B.
0 1 0 1	$n_3n_2n_3n_0$	Selects the nth display register character of Display Register B with the display outputs continuing to output the contents of Display Registers A and B.
0 1 1 0	$n_3n_2n_1n_0$	Selects the nth display register character of Display Register A and blanks the A and B display output (with hex 20).
0 1 1 1	$n_3n_2n_1n_0$	Selects the nth display register character of Display Register B and blanks the A and B display output (with hex 20).

WR1

Resets the internal display register pointer to Display Register A position 0 and forces the Display Registers to the blank code.

Note: A WR1 should follow a WR0 which changes the display mode.

WRM 1110 0000

Loads the contents of the display register location addressed by the internal display register pointer with the contents of ACC; then advances the displayed data by one digit in relation to the scan line timing, and increments the display register pointer. The display register pointer alternates between the A and B registers.

RDM 1110 1001

Loads ACC with the contents of the display register location addressed by the display register pointer and then increments the display register pointer. The display register pointer alternates between the A and B registers.

WMP 1110 0001

Loads the contents of the display register location addressed by the display register pointer with the contents of ACC.

RD3 1110 1111

Loads ACC with the contents of the display register location pointed to by the display register pointer.

NOTES:

1. The alternation pattern of the display register pointer is Display Register A position 0, Display Register B position 0, Display Register A position 1, etc.
2. The upper two (four) gas discharge characters, 16-17 (16-19), can be addressed only by incrementing the internal display register pointer above 15 by a WRM or RDM instruction in 18 (20) character gas discharge mode. If the internal display register pointer has been incremented above 15, then these characters can be read or written by a RD3 or WMP instruction.
3. Successive WRM commands will shift the output data (see gas discharge display output format below) one character forward in relation to the reset pulse. This will cause a wraparound shift left on the self-scan display. Hence, starting at register 0 and loading the display RAM will give a right-justified display — MSD first.

	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0
	RST	CLK	D_5	D_4	D_3	D_2	D_1	D_0
BLANK CODE:	X	X	1	0	0	0	0	0

Figure 1. Gas Discharge Display Output Format.

4. RDM will not cause any display shifting. The read order is MSD to LSD with the MSD stored in display register 0.

Scanned Sensor Mode

The instructions which are used in the scanned sensor mode are described below:

Mnemonic	Instruction Code	
SRC	0010	RRR1

The contents of the register pair RRR are used to select the 4269. An SRC is interpreted for scanned sensor mode as follows:

RRR_{even}	RRR_{odd}	
$D_3D_2D_1D_0$	$D_3D_2D_1D_0$	
0 1 X X	$n_3n_2n_1$ X	n_3-n_1 indicates an 8-bit sensor group to be read.

WR2 1110 0110

Clears the FIFO/RAM logic and the INT line.

RD1 1110 1110

Loads into ACC the upper 4 bits of the 8-bit sensor RAM group previously addressed by an SRC instruction.

RD2 1110 1110

Loads into ACC the lower 4 bits of the 8-bit sensor RAM group previously addressed by an SRC instruction.

NOTES:

1. In this mode, the 4269 PKD will continuously input the 64 matrix intersections of the sensor into the FIFO/Sensor RAM, which is organized as a 64-bit RAM.
2. The INT line will become active (V_{DD1}) and remain active whenever at least one intersection remains a logical one in the Sensor RAM.
3. The sensor group number set by the SRC is loaded into the internal display register pointer. Display mode instructions which change the internal display register pointer thus change the sensor group address.

Scanned Keyboard and Encoded Keyboard Modes

The instructions which are used in the scanned keyboard and encoded keyboard modes are described below:

Mnemonic	Instruction Code	
SRC	0010	RRR1

The contents of the register pair RRR are used to select the 4269. An SRC is interpreted as follows for scanned and encoded keyboard modes:

RRR_{even}	RRR_{odd}	
$D_3D_2D_1D_0$	$D_3D_2D_1D_0$	
0 1 X X	X X X X	SRC used only to select 4269.

WR2 1110 0110

Clears FIFO/RAM logic, the status buffer, and the INT line.

RD1 1110 1101

Reads the first nibble of the current FIFO register position.

RD2 1110 1110

Reads the second nibble of the current FIFO register position. FIFO register position is incremented to the next position.

RD0 1110 1100

Loads ACC with the FIFO status.

Notes:

1. The 4-bit FIFO status contains the number of valid characters (0-8) in the keyboard FIFO. However, in the event of an overrun, i.e., more than 8 characters entered, the 4-bit status will be set to a value of 15. The first eight characters entered prior to the overrun character will remain in the FIFO until cleared.
2. When a character is entered in the FIFO, the INT output pin wil go to V_{DD1}. When a character is read, the INT will change from V_{DD1} to V_{SS} (open) and back to V_{DD1} until the FIFO has been emptied. If a ninth character is inputted to the PKD before one complete character has been removed, the overrun status will be set. This will cause the INT line to remain active (V_{DD1}) even after all characters have been accessed. Overrun status can only be cleared by a WR2 or WR3 command (although the first eight FIFO characters can be read). This condition allows the user to detect an overrun condition if it occurs between the time the status buffer is checked and the time all characters have been read. It should be noted that an RD2 must be initiated after an RD1 to advance to the next FIFO word even if the second nibble is not desired.
3. For a 16-key Keyboard, successive RD2 instructions will be adequate for inputting the key code.

DESIGN CONSIDERATIONS

Display Modes

General Remarks

Each Display A and Display B output is capable of driving one standard TTL load. This is done by using a V_{SS} = +5, V_{DD} = -10V and V_{DD1} = GND. The V_{DD1} pin allows the PKD to interface to a variety of commercially available display arrays via a specified circuit. Gas discharge, phosphorescent, LED, and incandescent displays can all be used with a 4269. The interface requirements are determined by the selected display device. Current into each of the Display A and Display B output lines should not exceed 1.6mA.

The two 16 x 4 Display Registers A and B provide information in hexadecimal positive logic conventions. Hence, a 0000, negative logic V_{SS} on the data bus, will be 0000 (positive logic V_{DD1}) at the A and B display output. (The above is equivalent to one level inversion between the data outputs of the PKD and the CPU accumulator.)

Individual, Scanned Display Mode

The digit selection is achieved by using the eight scan lines, S_0-S_7, and the display select line RS. The RS output is used to multiplex the eight scan strobes to give sixteen separate strobes for up to 16 digits of display.

It should be noted that the LSD output position of both Display Registers A and B is gated out coincidently with S_0 time of the scan register. Following digit positions are also coincident. This feature allows an interface to 8 x 8 or 16 x 8 displays. For the first eight display digit positions, the RS output is at open drain. The remaining eight of the 16 digit positions are output sequentially with RS at V_{SS}. Sufficient active on-time (V_{SS}) is allowed at the scan strobe line (S_0-S_7) to illuminate the displayed digit. Sufficient time is also allowed between segments to extinguish segment and prevent overlapped illumination. If the 8 digit mode is selected with the WR1 instruction, the LSD will be gated out every S_0 time – not every other time.

For an aesthetic display transition, the display register outputs can be placed into the blank mode (all outputs to V_{SS}) via an SRC during the loading of the display register. The outputs can then be unblanked via another SRC when the display register has been completely loaded.

Gas Discharge Modes (Self-Scan)

An approximate 100 μsec period, 50% duty cycle clock will be provided to the gas discharge display. A reset pulse – one clock period long – will be generated every 111th clock period for the 16/18 digit displays or every 139th clock period for 20 digit displays. Character periods are either seven clock periods long (for 16 or 20 character displays) or six clock periods long (for 18 character displays). For either case, character data is valid for the first five clock periods of the character period. Character 0 (left-most digit) starts upon the rising edge of the reset signal. The blank code is $A_1 = V_{SS}$ and A_0, B_3 - $B_0 = V_{DD1}$, with A_3 and A_2 providing reset and clock functions respectively. For the 18 character gas discharge display mode, the data outputs are blank for the 108th, 109th, and 110th clock periods.

Keyboard Modes

Scanned Sensor Mode

The sensor interface consists of two groups of eight lines, the scan strobe lines (S_0-S_7) and the return sense lines (R_0-R_7). Each scan strobe is used to enable eight return lines, giving 64 total sense strobes for each complete scan. When in the sensor mode, the two key rollover and debounce logic is inhibited. This allows multiple valid intersection connections to be inputted. The SHIFT and CONTROL inputs are ignored in this mode.

Each sensor intersection will have a RAM location reserved. The designer should group the sensors in common groups of 4. This mode is intended to be used to scan a matrix of electronic intersections or mechanical contacts. Debouncing is to be performed under software control. The INT line will remain active (V_{DD1}) whenever a valid intersection has been detected. The scan strobe cycle is the same pattern of a logical 1 (V_{SS}) shifted in a field of zeros.

The sense return lines are read out by RD1/RD2 instructions as shown in Figure 2.

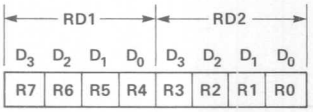


Figure 2. Sense Return.

Scanned Keyboard Mode

a. Key Depression Detection

These conditions can occur during the keyboard interrogation by the PKD (see timing diagram below).

1. Simultaneous Key Depression

   Two or more keys depressed within one complete single depression scan (approximately 11ms) is defined as a simultaneous key depression. If this condition occurs, the PKD continues to scan the keyboard and waits until one key remains depressed. It then treats the remaining key as a single key depression, as described below.

2. Single Key Depression

   When any single key (non-simultaneous) is depressed, an internal counter is started. The key code is also stored internally in a PKD temporary register with a code given by the values of the Scan and Return Lines. The PKD will then make four more complete scans of all keys. If no other keys are depressed during the fourth complete scan and the original key detected is still depressed at the end of the fourth scan, the key code is defined as a single key depression. The key code is then entered into the FIFO along with the value of the SHIFT and Control (S/C) input signals. If eight characters are already in the FIFO, the character will not be entered and the overrun will be set. When a character is entered in the FIFO, the INT line is activated to a logical "1" (V_{DD1}). If on the fourth complete scan the original key depressed is no longer depressed, the key is ignored as if it had never been depressed. This delay of four scan times, or approximately 11ms, thus provides the debounce function for the keyboard.

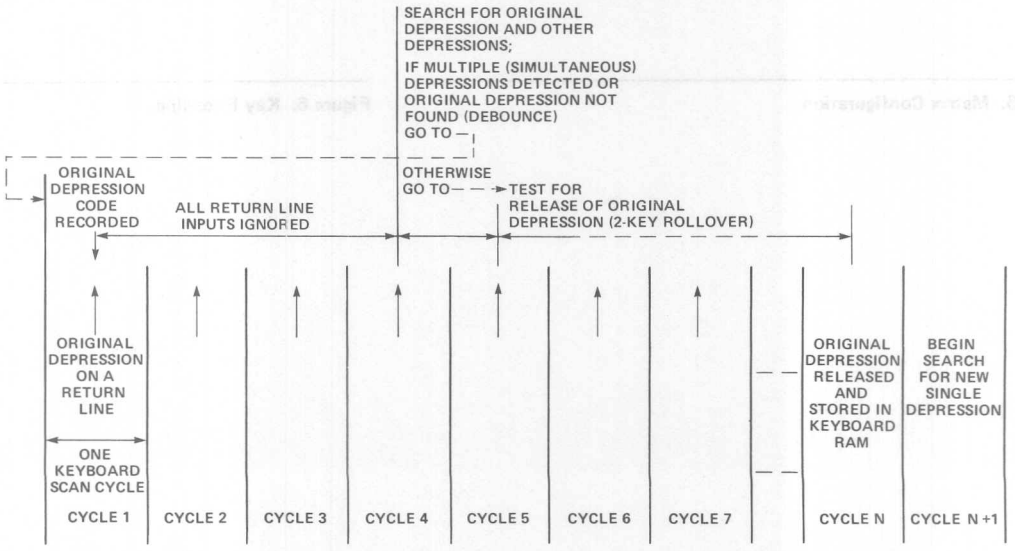


Figure 3. Keyboard Debounce and 2-Key Rollover Timing.

3. Two Key Rollover

The two key rollover operates as follows:

If a second key is depressed after a first key has been accepted by the PKD as a single key depression but the first key has not been released, then the second key will be treated as a new original depression after the first key has been released.

If a second key is depressed after a first key has been accepted by the PKD as a single key depression and the second key is released before the first key is released, the second key will be ignored.

b. Key Matrix Encoding

The keyboard matrix hardware configuration and associated matrix encoding is shown in Figures 4, 5, and 6.

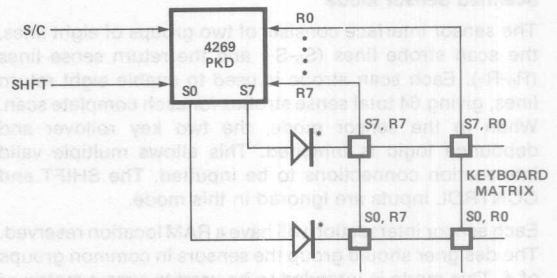


Figure 4. Hardware Configuration.

	R_0 000	R_1 001	R_2 010	R_3 011	R_4 100	R_5 101	R_6 110	R_7 111	SHIFT	S/C
S_0 000	0	1	2	3	4	5	6	7	X	X
S_1 001	8	9	10	11	12	13	14	15	X	X
S_2 010	16	17	18	19	20	21	22	23	X	X
S_3 011	24	25	26	27	28	29	30	31	X	X
S_4 100	32	33	34	35	36	37	38	39	X	X
S_5 101	40	41	42	43	44	45	46	47	X	X
S_6 110	48	49	50	51	52	53	54	55	X	X
S_7 111	56	57	58	59	60	61	62	63	X	X

Figure 5. Matrix Configuration.

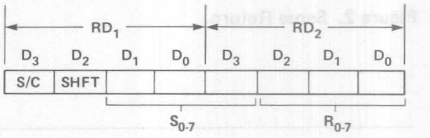


Figure 6. Key Encoding.

c. Expansion to 128 Key Scan

The basic mechanism of the PKD for scanning a 64 key matrix can be expanded to interface to a 128 key matrix. Note that the CONTROL (S/C) and SHIFT inputs cannot be used to directly encode 256 keys since the single key depression logic operates with the 6-bit matrix position code only. However, if full debounce and 2 key roll over control between two 64 key matrices is not necessary, then a configuration such as shown in Figure 7 may be used to add a seventh bit to the 6-bit matrix via the SHIFT or S/C input of the PKD. Alternately, two 4269 PKDs can be used for interfacing to the 128 keys.

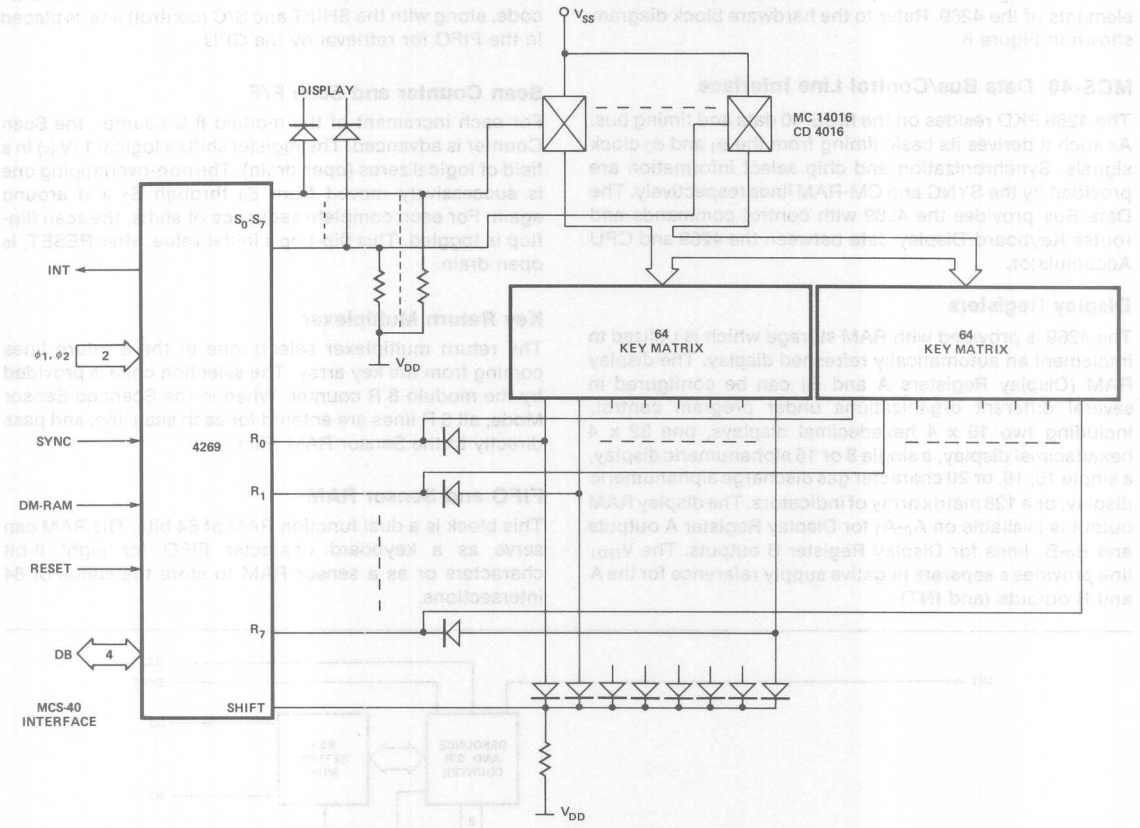


Figure 7. 128 Scanned Input Keys.

Encoded Keyboard Mode

Data Format

In the encoded keyboard mode, the eight return lines are directly loaded into the PKD's keyboard FIFO. For encoded keyboards using less than eight encoded bits, the remaining bits can be any desired signal, such as a multiplex signal between two keyboards or a special key flag.

HARDWARE DESCRIPTION

The following is a description of the major hardware elements of the 4269. Refer to the hardware block diagram shown in Figure 8.

MCS-40 Data Bus/Control Line Interface

The 4269 PKD resides on the MCS-40 data and timing bus. As such it derives its basic timing from the ϕ_1 and ϕ_2 clock signals. Synchronization and chip select information are provided by the SYNC and CM-RAM lines respectively. The Data Bus provides the 4269 with control commands and routes Keyboard/Display data between the 4269 and CPU Accumulator.

Display Registers

The 4269 is provided with RAM storage which is utilized to implement an automatically refreshed display. The display RAM (Display Registers A and B) can be configured in several different organizations under program control, including two 16 x 4 hexadecimal displays, one 32 x 4 hexadecimal display, a single 8 or 16 alphanumeric display, a single 16, 18, or 20 character gas discharge alphanumeric display, or a 128 matrix array of indicators. The display RAM output is available on A_0-A_3 for Display Register A outputs and B_0-B_3 lines for Display Register B outputs. The V_{DD1} line provides a separate negative supply reference for the A and B outputs (and INT).

S/R Counters and Debounce Logic

The S/R counters are two modulo 8 counters used to provide a unique 6-bit code for each of the 64 intersections provided by a matrix of eight Scan (S) Driver and eight Return (R) sense lines. The R counter is counted eight times for each S count. When keys, contacts, or controls are arranged in the matrix, each matrix intersection is examined for closure between the corresponding S and R line. If the 4269 is in the Scanned Keyboard Mode, an approximate 11 msec debounce time will be used to ascertain the validity of the connection. The valid 6-bit code, along with the SHIFT and S/C (control) line, is placed in the FIFO for retrieval by the CPU.

Scan Counter and Scan F/F

For each increment of the modulo 8 S counter, the Scan Counter is advanced. The register shifts a logical 1 (V_{SS}) in a field of logical zeros (open drain). The non-overlapping one is successively moved from S_0 through S_7 and around again. For each complete sequence of shifts, the scan flip-flop is toggled. This flip-flop's initial value, after RESET, is open drain.

Key Return Multiplexer

The return multiplexer selects one of the 8 return lines coming from the key array. The selection code is provided by the modulo 8 R counter. When in the Scanned Sensor Mode, all 8 R lines are entered for each scan line, and pass directly to the Sensor RAM (FIFO).

FIFO and Sensor RAM

This block is a dual function RAM of 64 bits. The RAM can serve as a keyboard character FIFO for eight 8-bit characters or as a sensor RAM to store the status of 64 intersections.

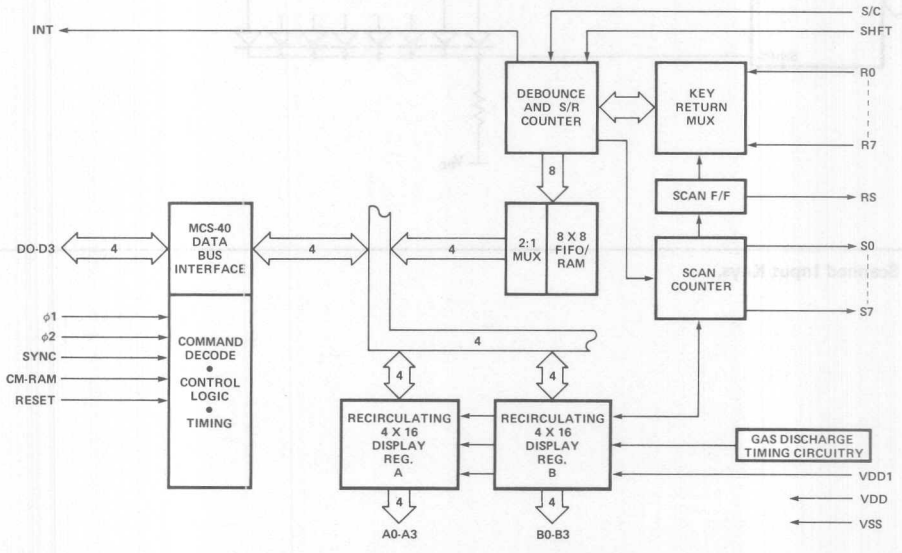


Figure 8. 4269 Hardware Block Diagram.

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0° to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400nsec; $t_{\phi D2}$ = 150nsec; Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless Otherwise Specified.

Symbol	Parameter	Limits			Unit	Test Condition
		Min.	Typ.	Max.		
I_{LI}	Input Leakage Current			10	µA	$V_{IL} = V_{DD}$
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
I_{LO}	Data Bus Output Leakage Current			10	µA	V_{OUT} = -12V
I_{OL}	Data Bus Sinking Current	8	15		mA	$V_{OUT} = V_{SS}$
I_{OL}	A_{0-3}/B_{0-3} Sinking Current		2.5		mA	$V_{DD1} = V_{SS}$ -5V, $V_{OUT} = V_{DD1}$ + .4V
I_{OL}	Interrupt Sinking Current		150		µA	$V_{OUT} = V_{DD1}$ +.5V
R_{OH}	Data Bus Output Resistance		150	250	Ω	$V_{OUT} = V_{SS}$ -.5V
R_{OH}	A_{0-3}/B_{0-3} Output Resistance		4		kΩ	$V_{OUT} = V_{SS}$ -2.6V
R_{OH}	S_{0-7} Output Resistance		250		Ω	$V_{OUT} = V_{SS}$ -1V
R_{OH}	RS Output Resistance		350		Ω	$V_{OUT} = V_{SS}$ -1V

A.C. Characteristics

$T_A = 0°C$ to $70°C$; $V_{SS} - V_{DD} = 15V \pm 5\%$.

Symbol	Parameter	Limits Min.	Typ.	Max.	Unit	Test Condition
t_{CY}	Clock Period	1.3		2	μsec	
$t_{\phi R}$	Clock Rise Time			50	nsec	
$t_{\phi F}$	Clock Fall Time			50	nsec	
$t_{\phi PW}$	Clock Width	380		480	nsec	
$t_{\phi D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	nsec	
$t_{\phi D2}$	Clock Delay ϕ_2 to ϕ_1	150			nsec	
t_W	Data-In, CM, SYNC Write Time	350	100		nsec	
t_H[1,2]	Data-In, CM, SYNC Hold Time	40	20		nsec	
t_{OS}[3]	Set Time (Reference)	0			nsec	
t_{ACC}	Data Bus Access Time			930	nsec	
t_{OH}	Data Bus Hold Time	50			nsec	
t_{RTSK}	Return Line Pull-Down Time		5		μs	C = 100pF; Scanned Keyboard Mode
t_{RTSN}	Return Line Pull-Down Time		30		μs	C = 100pF; Sensor Mode

Capacitance

Symbol	Parameter	Limits Min.	Typ.	Max.	Unit	Test Condition
C_ϕ	Clock Capacitance		8		pF	$V_{IN} = V_{SS}$
C_{DB}	Data Bus Capacitance		14	20	pF	$V_{IN} = V_{SS}$
C_{IN}	Input Capacitance			10	pF	$V_{IN} = V_{SS}$
C_{OUT}	Output Capacitance			10	pF	$V_{IN} = V_{SS}$

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.

2. All MCS-40 components which may transmit instruction on data to a 4004 or 4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.

3. t_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} in the same output access time referred to the leading edge of the next ϕ_2 clock pulse.

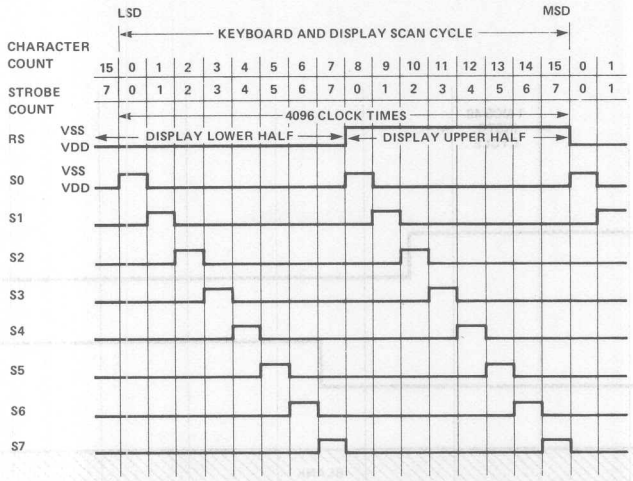


Figure 9. Individually Scanned Display Mode Timing.

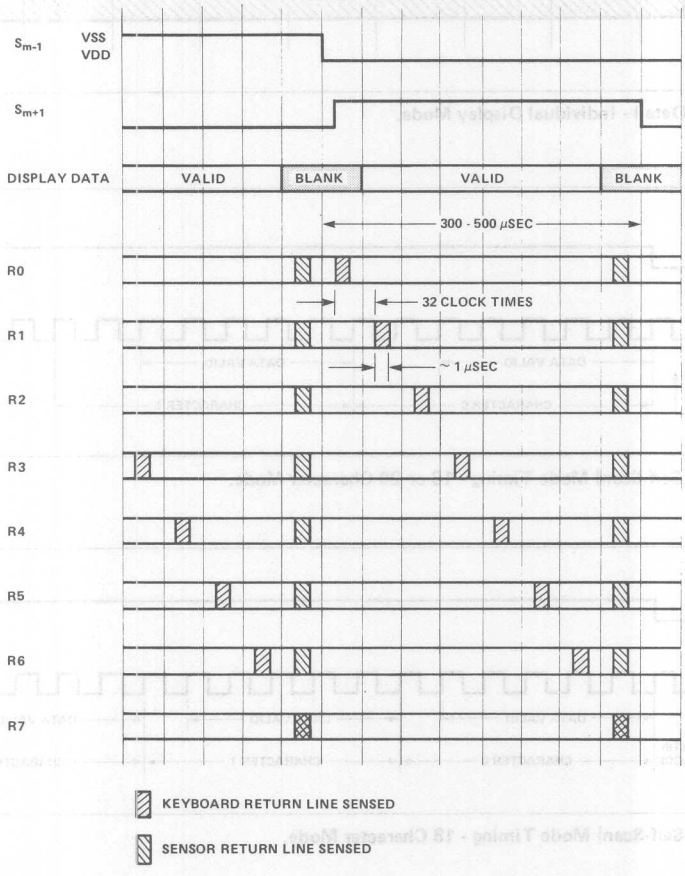


Figure 10. Detailed Timing of Strobe and Return Lines for Keyboard, Sensor, and Individual Scanned Display Modes.

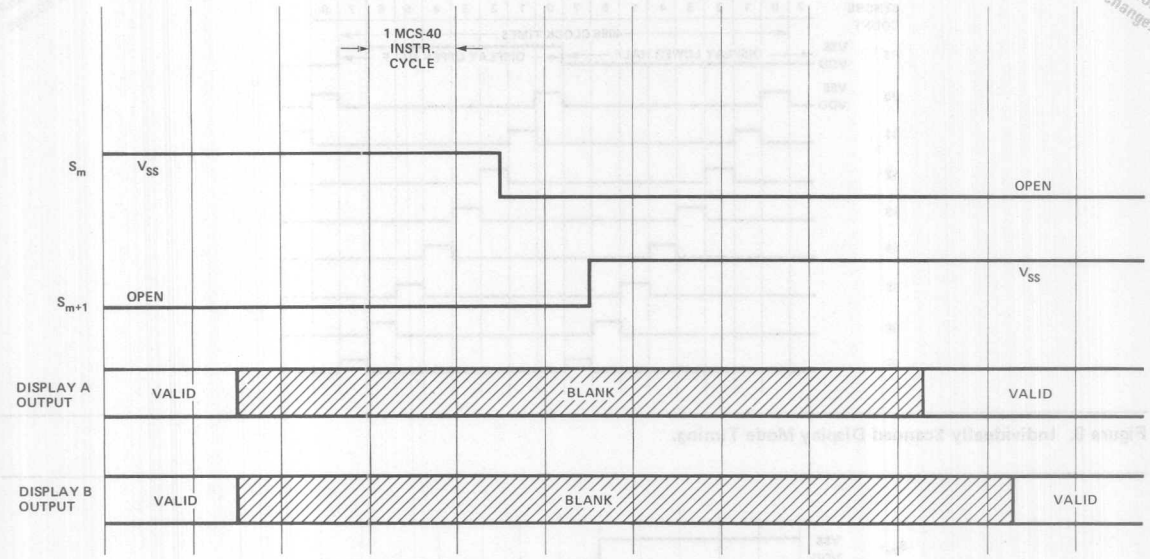


Figure 11. Data Blanking Detail - Individual Display Mode.

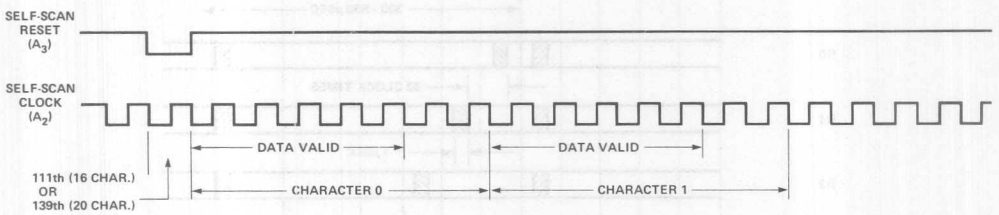


Figure 12. Gas Discharge (Self-Scan) Mode Timing - 16 or 20 Character Mode.

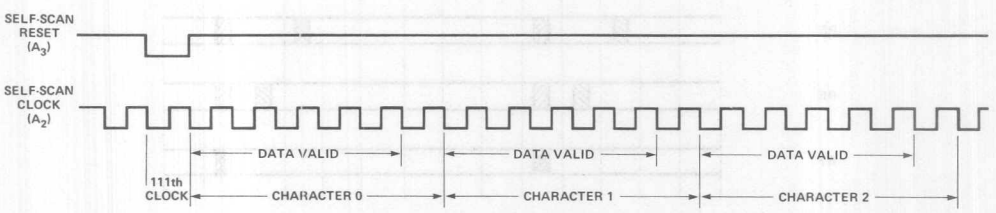


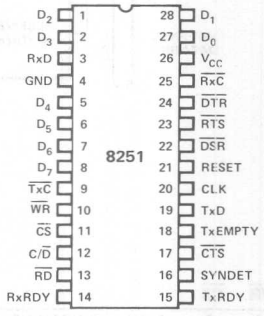
Figure 13. Gas Discharge (Self-Scan) Mode Timing - 18 Character Mode.

PROGRAMMABLE COMMUNICATION INTERFACE

- **Synchronous and Asynchronous Operation**
  - **Synchronous:**
    5-8 Bit Characters
    Internal or External Character Synchronization
    Automatic Sync Insertion
  - **Asynchronous:**
    5-8 Bit Characters
    Clock Rate — 1, 16 or 64 Times Baud Rate
    Break Character Generation
    1, 1½, or 2 Stop Bits
    False Start Bit Detection
- **Baud Rate —DC to 56k Baud (Sync Mode)**
  DC to 9.6k Baud (Async Mode)
- **Full Duplex, Double Buffered, Transmitter and Receiver**
- **Error Detection — Parity, Overrun, and Framing**
- **28-Pin DIP Package**
- **All Inputs and Outputs Are TTL Compatible**
- **Single 5 Volt Supply**
- **Single TTL Clock**

The 8251 is a Universal Synchronous/Asynchronous Receiver / Transmitter (USART) Chip designed for data communications in microcomputer systems. The USART is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM Bi-Sync). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPT. The chip is constructed using N-channel silicon gate technology.

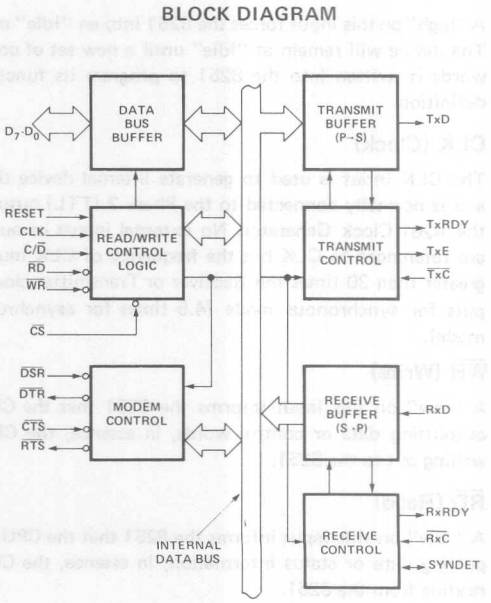
PIN CONFIGURATION



Pin Name	Pin Function
D_7-D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock
TxD	Transmitter Data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready (has character for CPU)
TxRDY	Transmitter Ready (ready for char. from CPU)

Pin Name	Pin Function
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET	Sync Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

BLOCK DIAGRAM



Functional Description

General

The 8251 Universal Synchronous/Asynchronous Receiver/Transmitter is programmed by the systems software for maximum flexibility. The 8251 can support virtually any serial data technique currently in use (including IBM "bi-sync").

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8251 to the system Data Bus. Data is transmitted or received by the buffer upon execution of input or output instructions. Control words, Command words and Status information are also transferred through the Data Bus Buffer.

Read/Write Control Logic

This functional block accepts inputs from the Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for device functional definition.

RESET (Reset)

A "high" on this input forces the 8251 into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251 to program its functional definition.

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the 4201 Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter clock inputs for synchronous mode (4.5 times for asynchronous mode).

$\overline{WR}$ (Write)

A "low" on this input informs the 8251 that the CPU is outputting data or control words, in essence, the CPU is writing out to the 8251.

$\overline{RD}$ (Read)

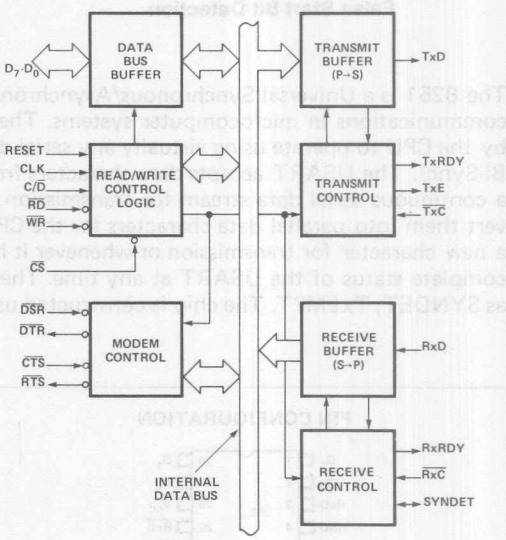
A "low" on this input informs the 8251 that the CPU is inputting data or status information, in essence, the CPU is reading from the 8251.

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251 that the word on the Data Bus is either a data character, control word or status information.
1 = CONTROL 0 = DATA

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8251. No reading or writing will occur unless the device is selected.



C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251 ⇒ DATA BUS
0	1	0	0	DATA BUS ⇒ 8251
1	0	1	0	STATUS ⇒ DATA BUS
1	1	0	0	DATA BUS ⇒ CONTROL
X	X	X	1	DATA BUS ⇒ 3-STATE

Modem Control

The 8251 has a set of control inputs and outputs that can be used to simplify the interface to almost any Modem. The modem control signals are general purpose in nature and can be used for functions other than Modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is general purpose in nature. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test Modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is general purpose in nature. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for Modem control such as Data Terminal Ready or Rate Select.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is general purpose in nature. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for Modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "low" on this input enables the 8251 to transmit data (serial) if the Tx EN bit in the Command byte is set to a "one."

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Transmitter Control

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

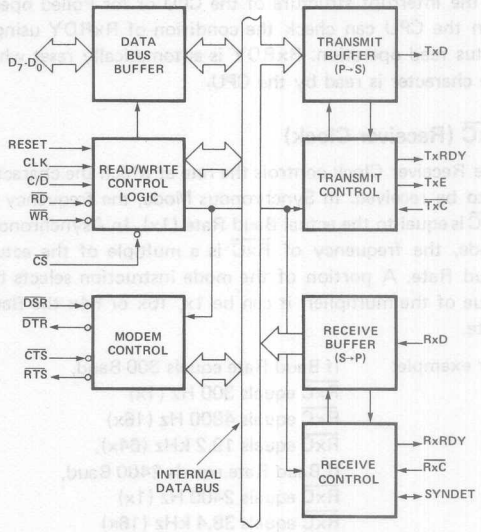
TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. It can be used as an interrupt to the system or for the Polled operation the CPU can check TxRDY using a status read operation. TxRDY is automatically reset when a character is loaded from the CPU.

TxE (Transmitter Empty)

When the 8251 has no characters to transmit, the TxE output will go "high". It resets automatically upon receiving a character from the CPU. TxE can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplexed operational mode.

In SYNChronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be transmitted automatically as "fillers".



$\overline{TxC}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the frequency of $\overline{TxC}$ is equal to the actual Baud Rate (1X). In Asynchronous transmission mode, the frequency of $\overline{TxC}$ is a multiple of the actual Baud Rate. A portion of the mode instruction selects the value of the multiplier; it can be 1x, 16x or 64x the Baud Rate.

For example:

If Baud Rate equals 110 Baud,
$\overline{TxC}$ equals 110 Hz (1x)
$\overline{TxC}$ equals 1.76 kHz (16x)
$\overline{TxC}$ equals 7.04 kHz (64x).
If Baud Rate equals 9600 Baud,
$\overline{TxC}$ equals 614.4 kHz (64x).

The falling edge of $\overline{TxC}$ shifts the serial data out of the 8251.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to the RxD pin.

Receiver Control

This functional block manages all receiver-related activities.

RxRDY (Receiver Ready)

This output indicates that the 8251 contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or for Polled operation the CPU can check the condition of RxRDY using a status read operation. RxRDY is automatically reset when the character is read by the CPU.

$\overline{\text{RxC}}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the frequency of $\overline{\text{RxC}}$ is equal to the actual Baud Rate (1x). In Asynchronous Mode, the frequency of $\overline{\text{RxC}}$ is a multiple of the actual Baud Rate. A portion of the mode instruction selects the value of the multiplier; it can be 1x, 16x or 64x the Baud Rate.

For example:
If Baud Rate equals 300 Baud,
$\overline{\text{RxC}}$ equals 300 Hz (1x)
$\overline{\text{RxC}}$ equals 4800 Hz (16x)
$\overline{\text{RxC}}$ equals 19.2 kHz (64x).
If Baud Rate equals 2400 Baud,
$\overline{\text{RxC}}$ equals 2400 Hz (1x)
$\overline{\text{RxC}}$ equals 38.4 kHz (16x)
$\overline{\text{RxC}}$ equals 153.6 kHz (64x).

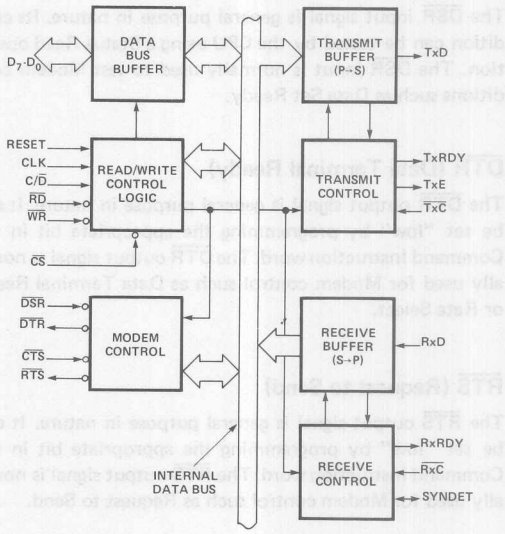
Data is sampled into the 8251 on the rising edge of $\overline{\text{RxC}}$.

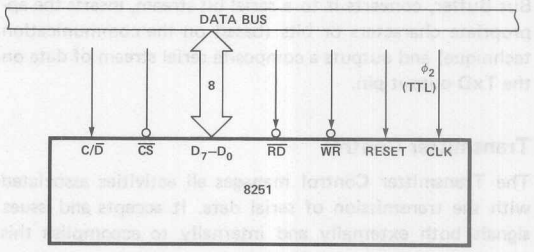
NOTE: In most communications systems, the 8251 will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{\text{TxC}}$ and $\overline{\text{RxC}}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

SYNDET (SYNC Detect)

This pin is used in SYNChronous Mode only. It is used as either input or output, programmable through the Control Word. It is reset to "low" upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251 has located the SYNC character in the Receive mode. If the 8251 is programmed to use double Sync characters (bi-sync), then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input, (external SYNC detect mode), a positive going signal will cause the 8251 to start assembling data characters on the falling edge of the next $\overline{\text{RxC}}$. Once in SYNC, the "high" input signal can be removed. The duration of the high signal should be at least equal to the period of $\overline{\text{RxC}}$.





8251 Interface

Operation Description

General

The complete functional definition of the 8251 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize the 8251 to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD PARITY etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251 is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251 is ready to receive a character. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251. The 8251 also receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "high" to signal the CPU that the 8251 has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU read operation.

The 8251 cannot begin transmission until the TxEN (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send (CTS) input. The TxD output will be held in the marking state upon Reset.

Programming the 8251

Prior to starting data transmission or reception, the 8251 must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251 and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

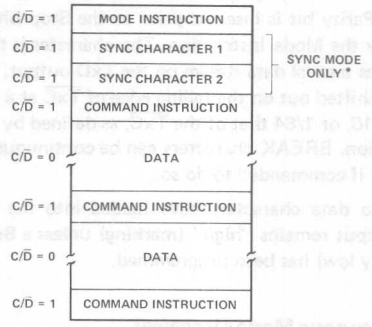
This format defines the general operational characteristics of the 8251. It **must** follow a Reset operation (internal or external). Once the Mode instruction has been written into the 8251 by the CPU, SYNC characters or Command instructions may be inserted.

Command Instruction

This format defines a status word that is used to control the actual operation of the 8251.

Both the Mode and Command instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251 for data communication.

All control words written into the 8251 after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251 at any time in the data block during the operation of the 8251. To return to the Mode Instruction format a bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251 back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.



*The second SYNC character is skipped if MODE instruction has programmed the 8251 to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the 8251 to ASYNC mode.

Typical Data Block

Mode Instruction Definition

The 8251 can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251, the designer can best view the device as two separate components sharing the same package. One is Asynchronous and the other Synchronous. The format definition can be changed "on the fly" but for explanation purposes the two formats will be isolated.

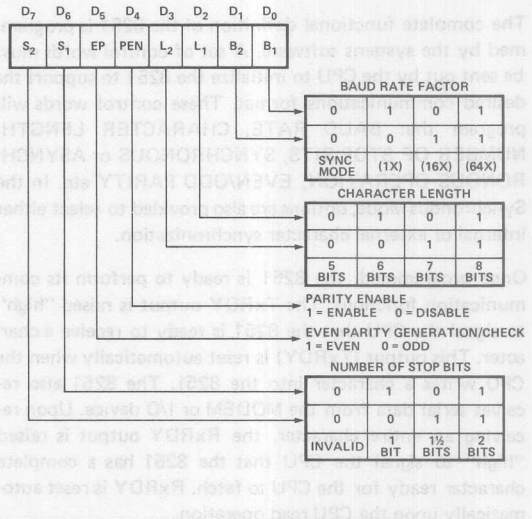
Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251 automatically adds a Start bit (low level) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

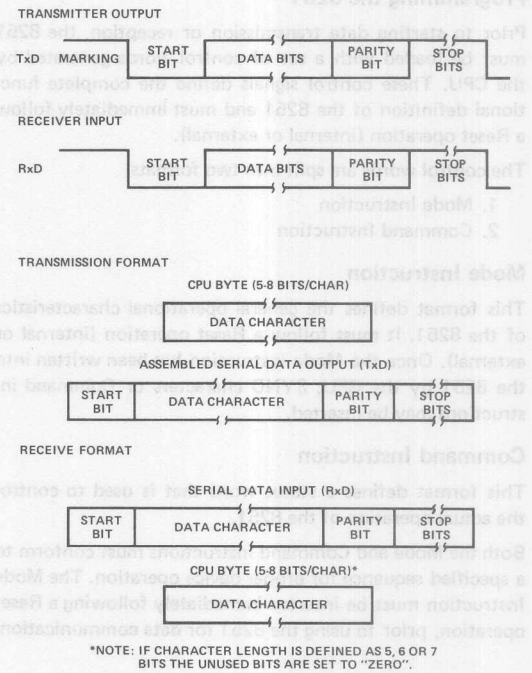
When no data characters have loaded into the 8251 the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If a parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251. The RxRDY signal is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the 8251.



Mode Instruction Format, Asynchronous Mode



Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251 which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251 with a character before the 8251 becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251 is empty and SYNC characters are being sent out. The TxEMPTY pin is internally reset by the next character being written into the 8251.

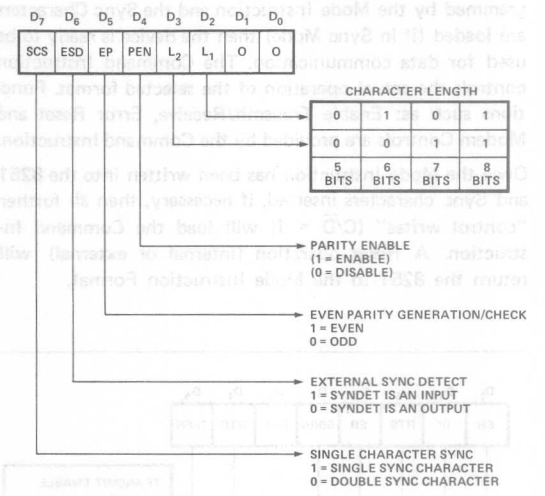
Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has been programmed, the receiver starts in a HUNT mode. Data on the RxD pin is then sampled in on the rising edge of $\overline{RxC}$. The content of the Rx buffer is continuously compared with the first SYNC character until a match occurs. If the 8251 has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ.

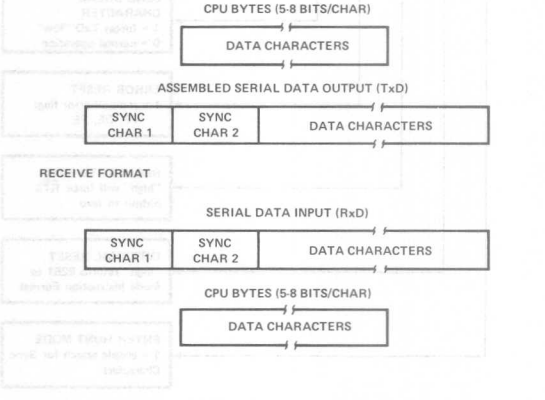
In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin. The high level can be removed after one $\overline{RxC}$ cycle.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.



Mode Instruction Format, Synchronous Mode

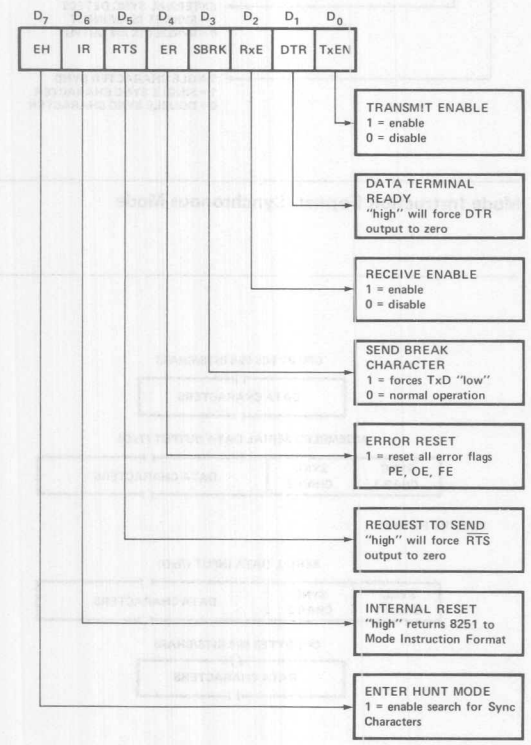


Synchronous Mode, Transmission Format

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251 has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251 and Sync characters inserted, if necessary, then all further "control writes" ($C/\overline{D}$ = 1) will load the Command Instruction. A Reset operation (internal or external) will return the 8251 to the Mode Instruction Format.



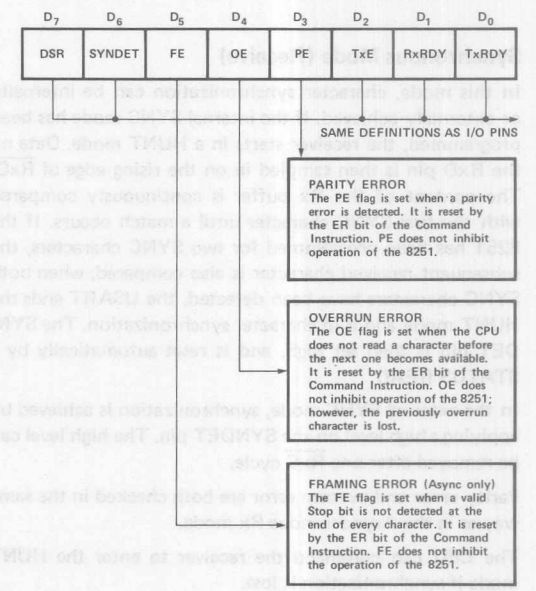
Command Instruction Format

STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251 has facilities that allow the programmer to "read" the status of the device at any time during the functional operation.

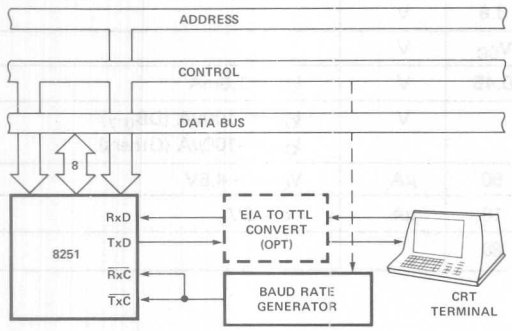
A normal "read" command is issued by the CPU with the C/D input at one to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251 can be used in a completely Polled environment or in an interrupt driven environment.

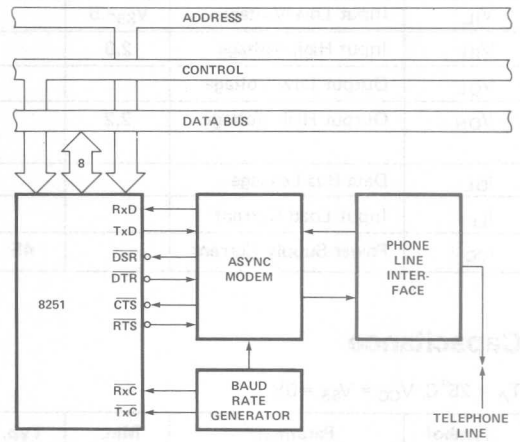


Status Read Format

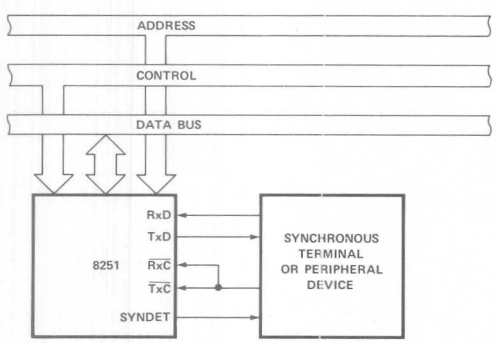
APPLICATIONS OF THE 8251



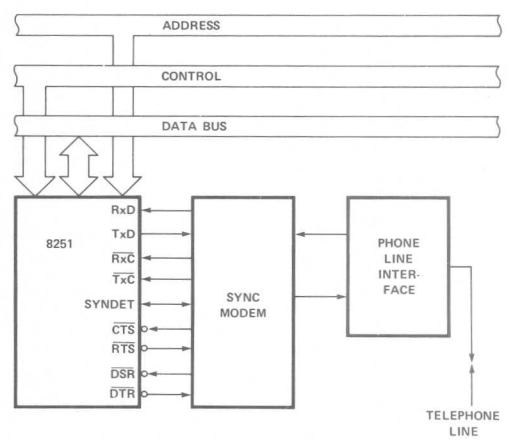
Asynchronous Serial Interface to CRT Terminal, DC-9600 Baud



Asynchronous Interface to Telephone Lines



Synchronous Interface to Terminal or Peripheral Device



Synchronous Interface to Telephone Lines

D.C. Characteristics

T_A = 0°C to 70°C; V_{CC} = 5.0V ± 5%; V_{SS} = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	V_{SS}-.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 1.6mA
V_{OH}	Output High Voltage	2.2			V	I_{OH} = -100μA (DB_{0-7}) I_{OH} = -100μA (Others)
I_{DL}	Data Bus Leakage			50	μA	V_{OUT} = 4.5V
I_{LI}	Input Load Current			10	μA	@ 5.5V
I_{CC}	Power Supply Current		45	80		

Capacitance

T_A = 25°C; V_{CC} = V_{SS} = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}.

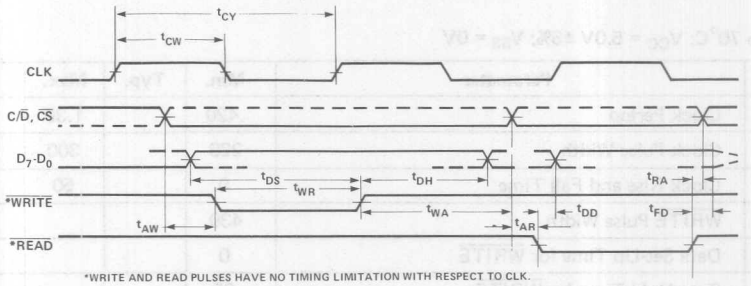
A.C. Characteristics

$T_A = 0°C$ to $70°C$; $V_{CC} = 5.0V \pm 5\%$; $V_{SS} = 0V$

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
t_{CY}	Clock Period	.420		1.35	μs	
$t_{\phi W}$	Clock Pulse Width	220		300	ns	
t_R, t_F	Clock Rise and Fall Time	0		50	ns	
t_{WR}	$\overline{WRITE}$ Pulse Width	430			ns	
t_{DS}	Data Set-Up Time for $\overline{WRITE}$	0			ns	
t_{DH}	Data Hold Time for $\overline{WRITE}$	65			ns	
t_{AW}	Address Stable before $\overline{WRITE}$	20			ns	
t_{WA}	Address Hold Time for $\overline{WRITE}$	35			ns	
t_{RD}	READ Pulse Width	430			ns	
t_{DD}	Data Delay from $\overline{READ}$	350			ns	C_L=100pF
t_{DF}	$\overline{READ}$ to Data Floating	150			ns	C_L=100pF
t_{AR1}	Address Stable before $\overline{READ}$, CE (C/D)	50			ns	
t_{RA1}	Address Hold Time for $\overline{READ}$, CE	5			ns	
t_{RA2}	Address Hold Time for $\overline{READ}$, C/D	370			ns	
t_{DTx}	TxD Delay from Falling Edge of TxC	1			μs	C_L=100pF
t_{SRx}	Rx Data Set-Up Time to Sampling Pulse	2			μs	C_L=100pF
t_{HRx}	Rx Data Hold Time to Sampling Pulse	2			μs	C_L=100pF
f_{Tx}	Transmitter Input Clock Frequency					
	1X Baud Rate	DC		56	KHz	
	16X and 64X Baud Rate	DC		615	KHz	
f_{Rx}	Receiver Input Clock Frequency					
	1X Baud Rate	DC		56	KHz	
	16X and 64X Baud Rate	DC		615	KHz	
t_{Tx}	TxRDY Delay from Center of Data Bit			16	CLK Period	C_L=50pF
t_{Rx}	RxRDY Delay from Center of Data Bit	15		20	CLK Period	
t_{IS}	Internal Syndet Delay from Center of Data Bit	20		25	CLK Period	
t_{ES}	External Syndet Set-Up Time before Falling Edge of RxC			15	CLK Period	

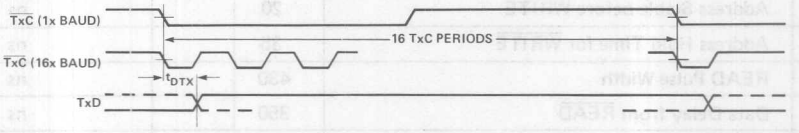
Note: The TxC and RxC frequencies have the following limitation with respect to CLK.
For ASYNC Mode, t_{Tx} or $t_{Rx} \geq 4.5\ t_{CY}$
For SYNC Mode, t_{Tx} or $t_{Rx} \geq 30\ t_{CY}$

READ AND WRITE TIMING

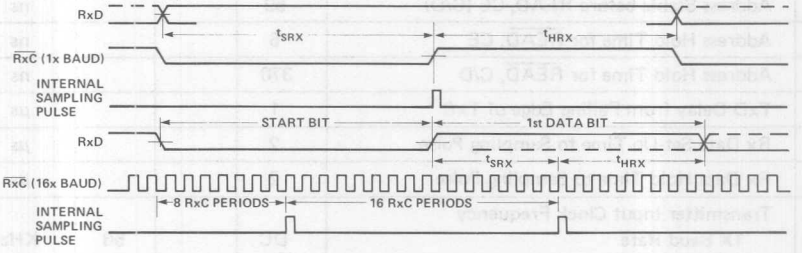


*WRITE AND READ PULSES HAVE NO TIMING LIMITATION WITH RESPECT TO CLK.

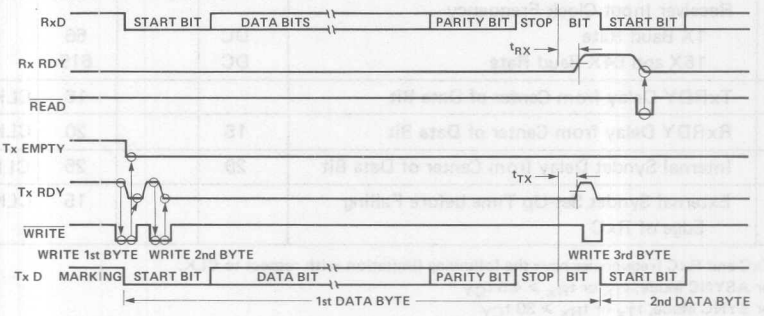
TRANSMITTER CLOCK AND DATA



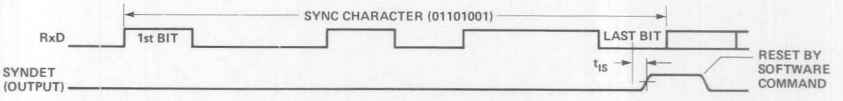
RECEIVER CLOCK AND DATA



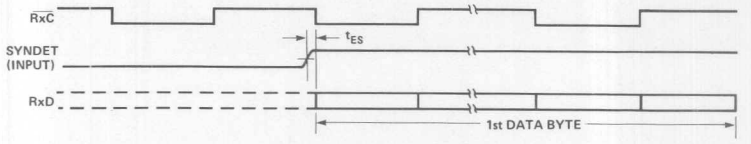
Tx RDY AND Rx RDY TIMING (ASYNC MODE)



INTERNAL SYNC DETECT



EXTERNAL SYNC DETECT



8253

PROGRAMMABLE INTERVAL TIMER

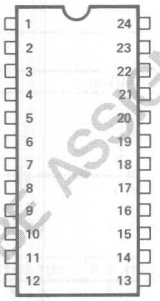
(Available 2nd Quarter 1976)

- **3 Independent 16-Bit Counters**
- **DC to 3 MHz**
- **Programmable Counter Modes**
- **Count Binary or BCD**
- **Single +5V Supply**
- **24 Pin Dual-in-line Package**

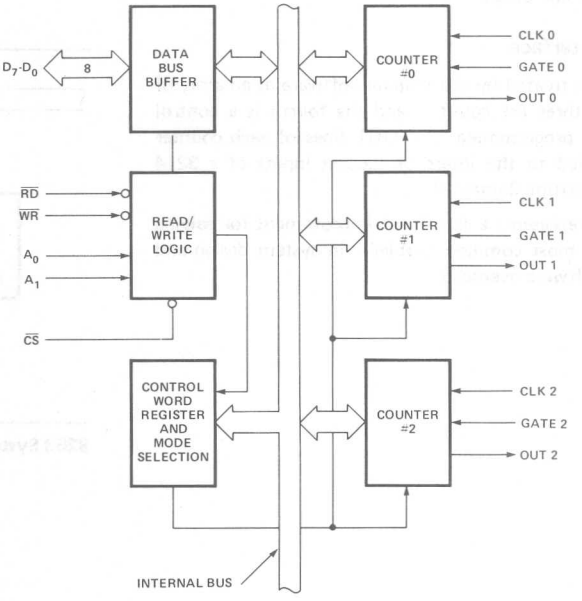
The 8253 is a programmable counter/timer chip designed for use as a microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as three independent 16-bit counters, each with a count rate from 0Hz to 3MHz. All modes of operation are software programmable.

PIN CONFIGURATION



BLOCK DIAGRAM



Functional Description

In Microcomputer-based systems the most common interface is to a mechanical device such as a printer head or stepper motor. All such devices have inherent delays that must be accounted for if accurate and reliable performance is to be achieved. The systems software allows for such delays by programmed timing loops. This type of programming requires significant overhead and maintenance of multiple loops gets extremely complicated.

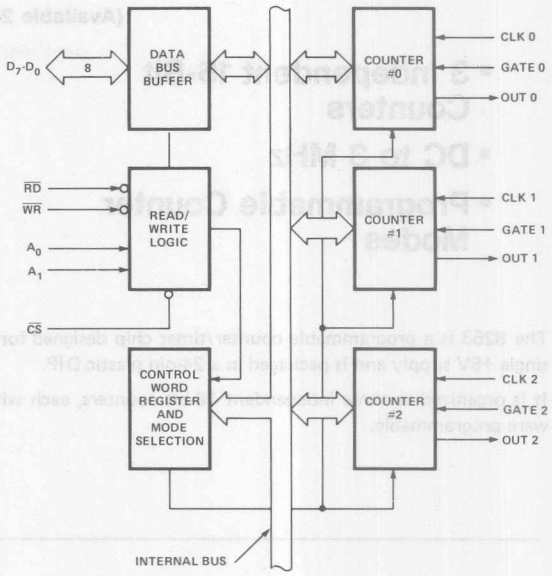
The 8253 Programmable Interval Timer is a single chip solution to system timing problems. In essence, it is a group of three 16-bit counters that are independent in nature but driven commonly as I/O peripheral ports. Instead of setting up timing loops in the system software, the programmer configures the 8253 to match his requirements. The programmer initializes one of the three counters of the 8253 with the quantity and mode desired then, upon command, the 8253 will count out the delay and interrupt the microcomputer when it has finished its task. It is easy to see that the software overhead is minimal and that multiple delays can be easily maintained by assigning interrupt levels to different counters. Other functions that are non-delay in nature and require counters can also be implemented with the 8253.

- Programmable Baud Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock

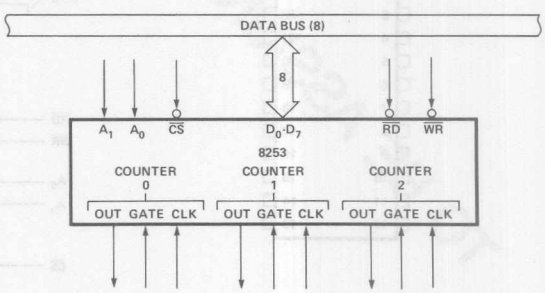
System Interface

The 8253 is treated by the systems software as an array of I/O ports; three are counters and the fourth is a control register for programming. The OUT lines of each counter could be tied to the interrupt request inputs of a 3214 Priority Interrupt Control Unit.

The 8253 represents a significant improvement for solving one of the most common problems in system design and reducing software overhead.



8253 Block Diagram.

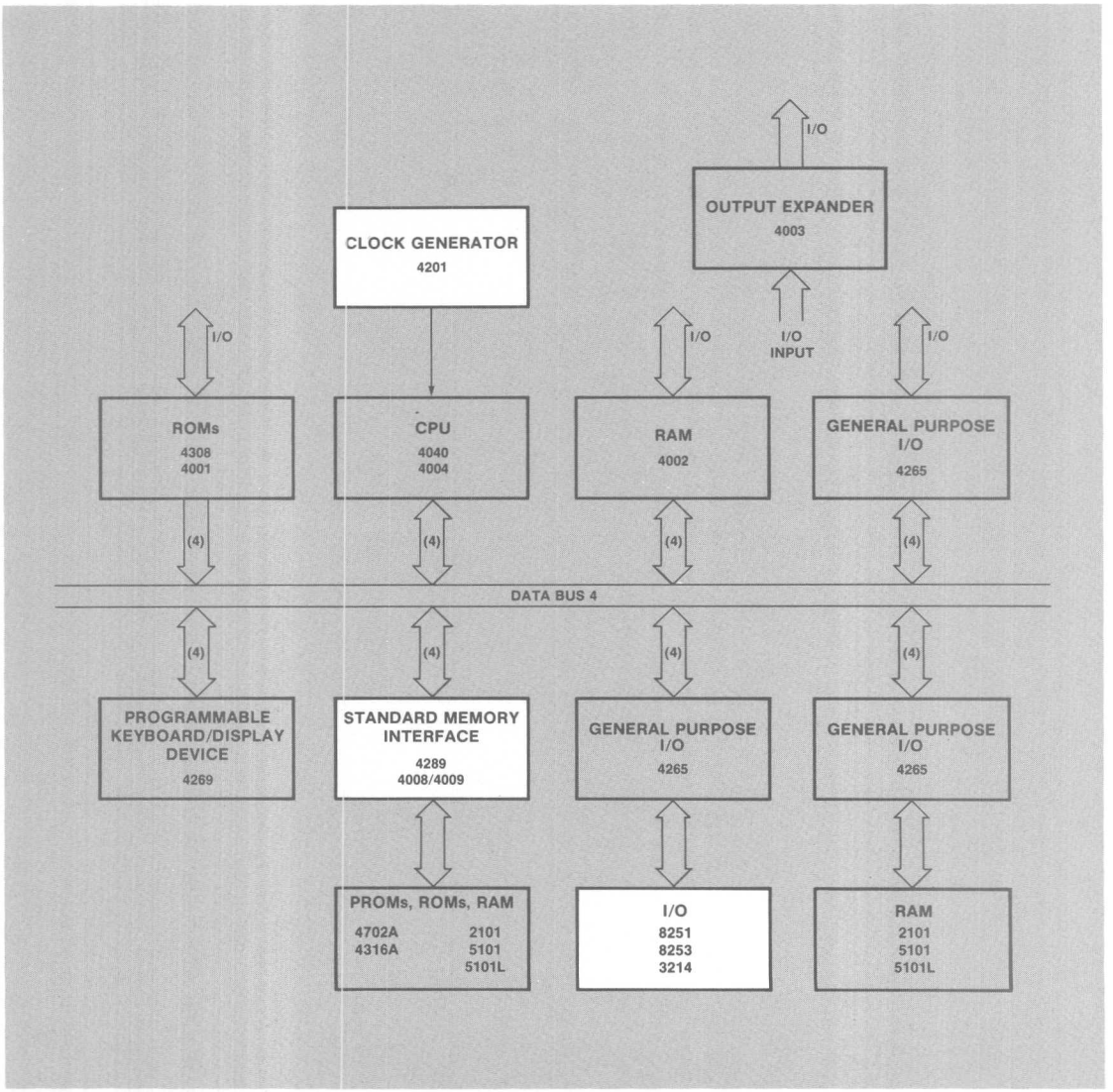


8253 System Interface.

intel® Microcomputer Systems

PERIPHERALS

4201
4289
4008/4009
3205
3216/3226
3214



PERIPHERALS

4201	3205
4289	3216/3226
4008/4009	3214

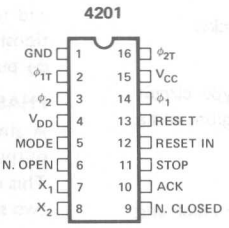
4201
CLOCK GENERATOR

- **Complete Clock Requirements for MCS-40™ Systems**
- **Crystal Controlled Oscillator (XTAL External)**
- **MOS and TTL Level Clock Outputs**
- **Provides MCS-40 Reset Function Signal**
- **Standard Operating Temperature Range of 0° to 70° C**
- **Also Available with -40° to +85° C Operating Range**

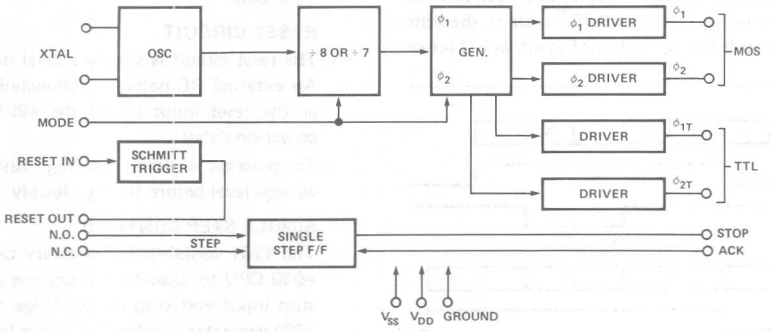
The 4201 is a CMOS integrated circuit designed to fill the clock requirements of the MCS-40 microcomputer family. The 4201 contains a crystal controlled oscillator (XTAL external), clock generation circuitry, and both MOS and TTL level clock driver circuits.

The 4201 also performs the power on reset function required by MCS-40 components and provides the logic necessary to implement the single-step function of the 4040 central processor unit.

PIN CONFIGURATION



BLOCK DIAGRAM



Pin Description

Pin No.	Designation	Description of Function
1	GND	Circuit ground potential. This pin can be left floating for low power application. MOS clock output will be operative, TTL clock outputs will not.
2	ϕ1T	Phase 1 TTL level clock output. Positive true.
3	ϕ2	Phase 2 MOS level clock output. Directly drives all MCS-40 components.
4	V_{DD}	Main Power Supply Pin. $V_{DD} = V_{CC}$ -15V ±5%.
5	MODE	Counter mode control pin. Determines whether counter divides basic frequency by 8 or 7. Mode 1 = $V_{CC} \Rightarrow \div 7$ Mode 2 = $V_{DD} \Rightarrow \div 8$
6	N. OPEN	Input of single step circuitry to which normally open contact of SPDT switch is connected.
7	X1	External Crystal Connection. This pin may be driven by an external frequency source. X2 should be left unconnected.
8	X2	External Crystal Connection.
9	N. CLOSED	Input of single step circuitry to which normally closed contact of SPDT switch is connected.
10	ACK	Acknowledge input to single step circuitry normally connected to stop acknowledge output of 4040.
11	STOP	Stop output of single step circuitry normally connected to stop input of 4040. A SPDT toggle switch may be inserted in this line for RUN/HALT control.
12	RESET IN	Input to which RC network is connected to provide power-on reset timing.
13	RESET	Reset signal output which directly connects to all MCS-40 reset inputs. This signal is active low.
14	ϕ1	Phase 1 MOS level clock output. Directly drives all MCS-40 clock inputs.
15	V_{CC}	Circuit reference potential – most positive supply voltage.
16	ϕ2T	Phase 2 TTL level clock output. Positive true.

Functional Description

The 4201 consists of the following functional blocks:

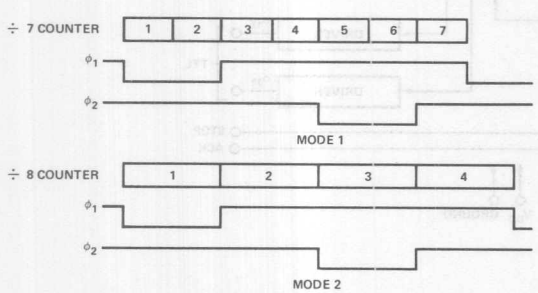
CRYSTAL OSCILLATOR

The oscillator is a simple series mode crystal-type circuit consisting of two inverters biased in the active region, and a series crystal element.

PROGRAMMABLE SHIFT REGISTER

The shift register in the 4201 divides the master clock and generates the proper states for generating the desired two-phase clock. The circuit is a seven bit dynamic device which circulates a logical "1" through a field of zeroes. The output of the various states are then combined to provide the proper clock waveforms. This provides a divide by 7 function.

In order to maintain the proper clock timing over the full operating frequency range of the MCS-40™ system, the shift register is programmable (using mode pin) as either a 7 bit or 4 bit device. When in the 4 bit mode the clock is divided by 2 and then by 4 to provide a divide by 8 function. The relationship between the phases is equal; that is, ϕ_1 pulse width, ϕ_2 pulse width, ϕ_1 to ϕ_2 and ϕ_2 to ϕ_1 times are all equal.



4201 Shift Register Modes.

PHASE DECODER

A simple gate complex is used to decode the shift register outputs to provide phase 1 and phase 2 clock waveforms. This circuitry is controlled by the mode input to achieve the two sets of timing discussed in the previous section.

OUTPUT BUFFERS

There are two sets of output buffers for the 2 phase clock. One set is the MOS level drivers designed to directly drive a full complement of MSC-40 components. The second set provides TTL compatible outputs which can drive one standard TTL load.

RESET CIRCUIT

The reset circuit is simply a level detector and driver stage. An external RC network connected between V_{DD} and V_{SS} at the reset input pin of the 4201 provides the required power-on delay.

To generate a reset, the V_{DD} supply must reach its full voltage level before the V_{SS} supply turns on.

SINGLE STEP CONTROL

The 4201 contains the necessary circuitry for allowing the 4040 CPU to execute instructions one at a time. Using the stop input and stop acknowledge output of the 4040, the 4201 generates a pulse that allows the 4040 to perform only one instruction. The stop command can be provided by a SPDT pushbutton directly since debouncing is provided by the 4201. A SPST toggle switch, in series with the $\overline{STOP}$ line, provides the Run/Halt feature.

Absolute Maximum Ratings*

Storage Temperature . . . . . -55°C to 150°C Ambient
Operating Temperature . . . . . 0°C to 70°C Ambient
Maximum Positive Voltage . . . . . V_{CC} +.5V
Maximum Negative Voltage . . . . . V_{DD} -.3V
Maximum Power Dissipation 1.0W
Maximum Supply Voltage V_{CC} -V_{DD} . . . . . 17V[1]
Maximum Supply Voltage V_{CC} -V_{DD} . . . . . 17V[2]

Notes: 1. C_{LOAD}, ϕ_1 and $\phi_2 \geq 100$pF.
2. C_{LOAD}, ϕ_1 and $\phi_2 = 0$; R = 68Ω, V_{DD} Pin to V_{DD}; Bypass Capacitor at V_{DD} Pin.

*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.

D.C. Characteristics

$T_A = 0°C$ to 70°C; V_{CC}-V_{DD} = 15V ±5%; GND = V_{CC} -5V ±5%.

Symbol	Parameter	Limit		Units	Conditions
		Min.	Max.		
I_{LI}	Input Leakage Current		10	μA	$V_{IL} = V_{DD}$ All inputs except X_1, X_2, N. Open, N. Closed
V_{IH}	Input High Voltage	V_{CC}-1.5	V_{CC}+.5	V	All inputs except X_1, X_2, Reset
V_{IL}	Input Low Voltage	V_{DD}	V_{CC}-13	V	All inputs except X_1, X_2, Reset
V_{OL}	Output Low Voltage	V_{DD}	V_{CC}-13.4	V	Capacitance load only
V_{OH}	Output High Voltage	V_{CC}-1.5	V_{CC}	V	Capacitance load only
V_{OL}	ϕ_{1T}, ϕ_{2T}		GND+.5	V	I_{OL} = 1.6mA
V_{OH}	ϕ_{1T}, ϕ_{2T}	V_{CC}-.75		V	I_{OH} = -400μA
I_{OL}	ϕ_1, ϕ_2 Sink Current	400		mA	$V_{OUT} = V_{CC}$; Pulse Width ≤1μsec
I_{OL}	ϕ_{1T}, ϕ_{2T} Sink Current	15		mA	$V_{OUT} = V_{CC}$
I_{OL}	Reset Sink Current	6		mA	$V_{OUT} = V_{CC}$
I_{OL}	Stop Sink Current	1		mA	$V_{OUT} = V_{CC}$
I_{OH}	ϕ_1, ϕ_2 Source Current	180		mA	$V_{OUT} = V_{DD}$
I_{OH}	ϕ_{1T}, ϕ_{2T} Source Current	8		mA	$V_{OUT} = V_{DD}$
I_{OH}	Reset Source Current	6		mA	$V_{OUT} = V_{DD}$
I_{OH}	Stop Source Current	1		mA	$V_{OUT} = V_{DD}$
I_{DD}	Average Supply Current		20	mA	5.185MHz Crystal, C_{LOAD} ϕ_1 and ϕ_2 = 20pF
V_{IL}	Reset Input Low Voltage	V_{DD}	V_{CC}-11	V	
V_{IH}	Reset Input High Voltage	V_{CC}-6.5	V_{CC}+.5	V	
R_1	Pull Up Resistance on N. Open, N. Closed	20	120	KΩ	$V_{IN} = V_{DD}$

Capacitance

f = 1MHz; T_A = 25°C

Symbol	Parameter	Limit		Units	Conditions
		Min.	Max.		
C_{IN}	Input Capacitance		5	pF	All Inputs except X_1, X_2
C_{OUT}	ϕ_1, ϕ_2 Output Capacitance		40	pF	
C_{OUT}	ϕ_{1T}, ϕ_{2T} Output Capacitance		10	pF	
C_{OUT}	Stop Reset Output Capacitance		5	pF	

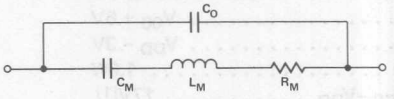
XTAL Specifications

Range: 3.5 - 5.185 MHz

Mode: Parallel Resonant

Recommended: Crystek 5.185 MHz
Spec. No. CY8A or CTS
Knights 4201-5.185 or Equivalent

CTS Knights
XTAL Equivalent Circuit



$C_O \approx$ 3-5pF
$C_M \approx$ 10fF
$R_M \approx 50\Omega$
$L_M \approx \frac{1}{(2\pi f)^2 C_M}$

XTAL Capacitance Requirements: 15-20 pF

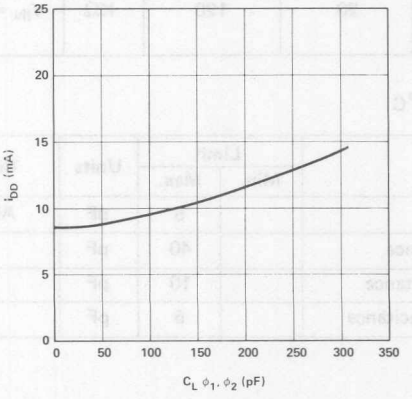
A.C. Characteristics $T_A = 0°C$ to $70°C$; $V_{CC} - V_{DD} = 15V \pm 5\%$; $G = V_{CC} - 5V \pm 5\%$

Symbol	Parameter	Limit			Units	Conditions
		Min.	Typ.	Max.		
t_{CY}	Clock Period		$t_{XTAL}*7$		ns	Mode = V_{CC}
$t_{\phi PW}$	Clock Pulse Width	$(2/7)t_{CY}-10$	$(2/7)t_{CY}$	$(2/7)t_{CY}+10$	ns	
$t_{\phi D1}$	Clock Delay from ϕ_1 to ϕ_2	$(2/7)t_{CY}-10$	$(2/7)t_{CY}$	$(2/7)t_{CY}+10$	ns	
$t_{\phi D2}$	Clock Delay from ϕ_2 to ϕ_1	$(1/7)t_{CY}-10$	$(1/7)t_{CY}$	$(1/7)t_{CY}+10$	ns	
t_{CY}	Clock Period		$t_{XTAL}*8$		ns	Mode = V_{DD}
$t_{\phi PW}$	Clock Pulse Width	$(1/4)t_{CY}-10$	$(1/4)t_{CY}$	$(1/4)t_{CY}+10$	ns	
$t_{\phi D1}$	Clock Delay from ϕ_1 to ϕ_2	$(1/4)t_{CY}-10$	$(1/4)t_{CY}$	$(1/4)t_{CY}+10$	ns	
$t_{\phi D2}$	Clock Delay from ϕ_2 to ϕ_1	$(1/4)t_{CY}-10$	$(1/4)t_{CY}$	$(1/4)t_{CY}+10$	ns	
$t_{\phi D3}$	TTL Clk to MOS Clk Skew[1]	0		40	ns	
$t_{\phi r}, t_{\phi f}$	Clock Rise and Fall Time			50	ns	C_L=300pF=ϕ_1,ϕ_2; C_L=50pF on ϕ_{1T},ϕ_{2T}
t_D	Delay from Acknowledge to Stop			1	μs	C_L=20pF

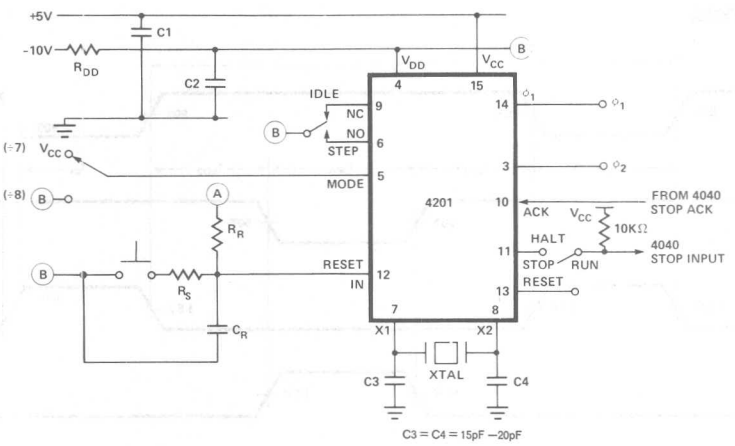
Note: 1. See waveforms section for phase relationships between ϕ_1, $\phi_1 T$, ϕ_2, and $\phi_2 T$.
2. Proper system operation of all members of the MCS-40 component family is guaranteed with the 4201 Clock Generator at 1.35 μsec $\leq t_{CY} \leq$ 2 μsec.

Typical Characteristics

I_{DD} CURRENT VS. LOAD CAPACITANCE



Clock Generator Implementation



Power Supply Voltages

The purpose of R_{DD} is both to limit ϕ_1 and ϕ_2 rise times and to drop V_{DD} at the 4201 pin. Values for R_{DD} as a function of ϕ_1, ϕ_2 load capacitance are:

For C_{LOAD} <50pF; use R_{DD} = 100Ω.
For 50pF <C_{LOAD} <100pF; use R_{DD} = 68Ω.
For 100pF <C_{LOAD} <300pF; use R_{DD} = 27Ω.
For C_{LOAD} >300pF; use R_{DD} = 10Ω.

All 4201 functions requiring the V_{DD} voltage should use the pin V_{DD} or node Ⓑ on the 4201 side of resistor R_{DD}.

Operation is guaranteed with V_{CC}-V_{DD} = 15V ±5%. During system power-up or during power supply glitching, the maximum magnitude of (V_{CC}-V_{DD}) must be limited to 17 volts.

With V_{CC} = +5V, V_{DD} = -10V, bypass capacitor C1 of 1 μF and C2 of .1 μF in parallel from V_{CC} to GND and V_{DD} to GND provide excellent bypassing.

Single-Supply Systems (+15V or -15V)

Recommended 4201 circuit modifications for single supply systems are:

1. The 1 μF capacitor C1 should be between V_{DD} and V_{CC}.
2. Other capacitors shown as being grounded should be connected to V_{CC}.
3. Reset R – C should be connected to V_{CC}.
4. The current limiting resistor R_{DD} is still needed in the V_{DD} line.

Crystals

Either ÷7 or ÷8 Modes may be used. Mode equals V_{CC} for ÷7, Mode equals V_{DD} for ÷8. The XTAL range should be between 500 kHz (4 MHz XTAL, ÷8 MODE) and 740 kHz (5.185 MHz XTAL, ÷7 MODE). These XTAL may be found as standard products from CTS Knights or Crystek.

The XTAL terminals, X1 and X2, should each be tied to 15pF-20pF capacitors C3 and C4 to GND. Exact values of C3 and C4 should be selected such that total capacitance seen at X_1 and X_2 inputs, including lead and board capacitance, allows proper oscillation start-up.

Reset Network

The Reset input has V_{IL} = V_{CC} -11 Volts and V_{IH} = V_{CC} -6.5 Volts, with about 1 Volt of hysteresis (Schmitt circuit).

Node Ⓐ must be tied to GND or V_{CC} = +5V; and R_R and C_R selected, such that the negative V_{DD} transition moves the Reset input below V_{IL}.

Tying node Ⓐ to GND and making C_R very large, i.e.>1μF, will allow the greatest freedom in V_{CC} and V_{DD} rise times during turn-on. Tying node Ⓐ to GND will also cause Reset after a V_{DD} glitch to GND.

The purpose of R_S at 510Ω or 1kΩ, is to limit Reset input fall time on manual Reset, so that the Reset input does not fall below V_{DD}.

TTL Clock Outputs

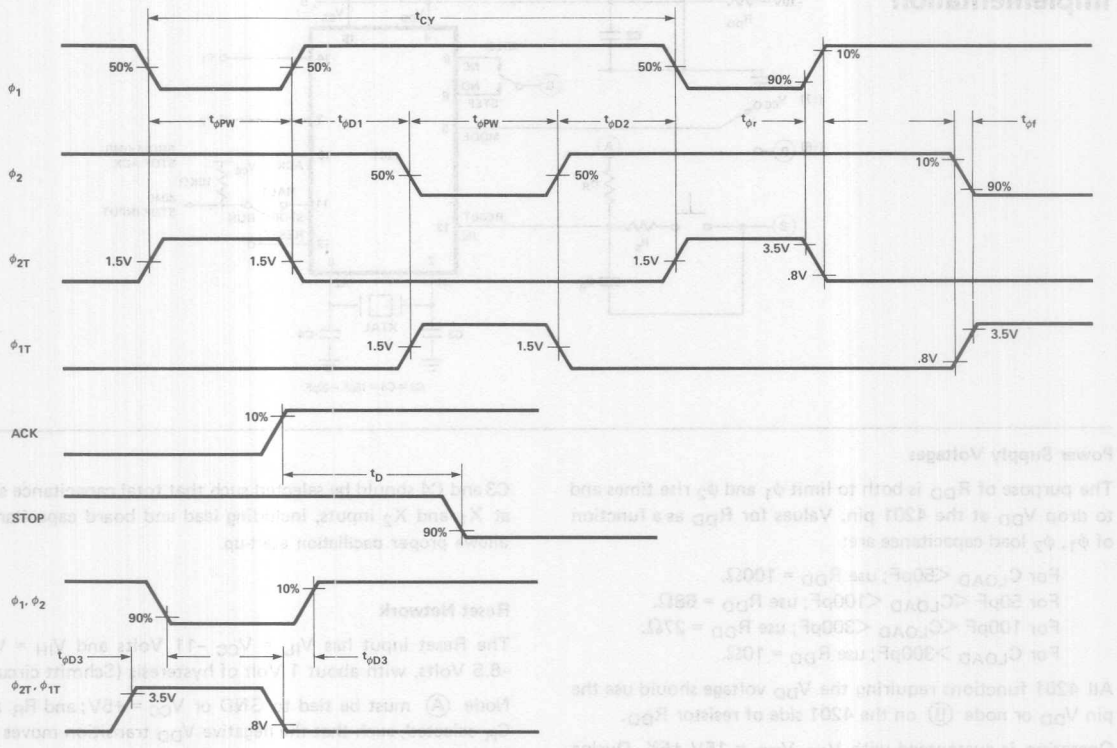
If ϕ_{1T} and ϕ_{2T} are used, GND pin should be tied to logic ground. ϕ_{1T} and ϕ_{2T} levels will be equal to V_{CC} and the GND pin level.

Unused Functions

If any of the 4201 functions listed below are not used, it is recommended that the pins be connected as described below:

1. ϕ_{1T}, ϕ_{2T} – Tie GND, ϕ_{1T}, ϕ_{2T} to V_{CC}.
2. Single Step – Tie NO to V_{CC}
   NC to Node Ⓑ (V_{DD} pin of 4201)
   STOP ACK to V_{CC}
   STOP left open
3. Reset – Tie RESET IN to V_{CC}
   RESET OUT to V_{CC}.

Waveforms


t_{CY}
ϕ_1
50%
50%
50%
90%
10%
$t_{\phi PW}$
$t_{\phi D1}$
$t_{\phi PW}$
$t_{\phi D2}$
$t_{\phi r}$
$t_{\phi f}$
ϕ_2
50%
50%
10%
90%
ϕ_{2T}
1.5V
1.5V
1.5V
3.5V
.8V
ϕ_{1T}
1.5V
1.5V
.8V
3.5V
ACK
10%
t_D
STOP
90%
ϕ_1, ϕ_2
10%
90%
$t_{\phi D3}$
$t_{\phi D3}$
ϕ_{2T}, ϕ_{1T}
3.5V
.8V

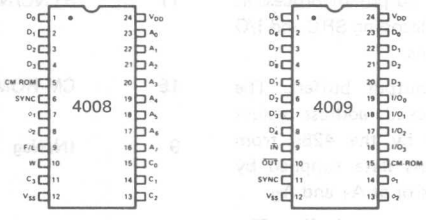
4008/4009

STANDARD MEMORY AND I/O INTERFACE SET

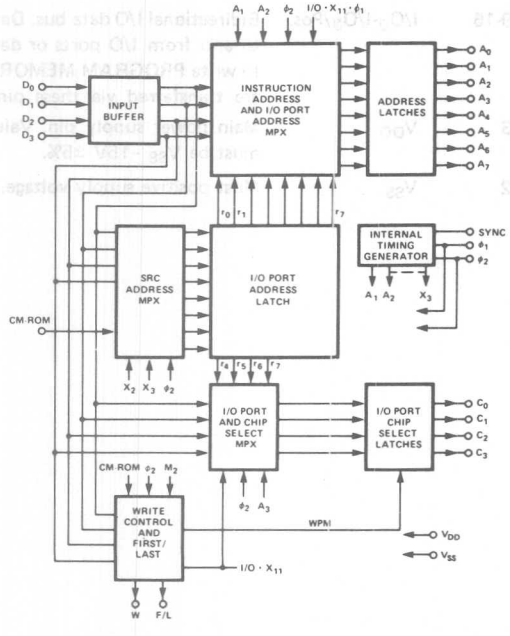
- **Direct Interface to Standard Memories**
- **Allows Write Program Memory**
- **24 Pin Dual In-Line Packages**
- **Standard Operating Temperature Range of 0° to 70 °C**

The standard memory and I/O interface set (4008/4009) provides the complete control functions performed by the 4001 or 4308 in MCS-40™ systems. The 4008/4009 are completely compatible with other members of the MCS-40 family. All activity is still under control of the CPU. One set of 4008/4009 and several TTL decoders is sufficient to interface to 4K words of program memory, sixteen four-bit input ports and sixteen four-bit output ports.

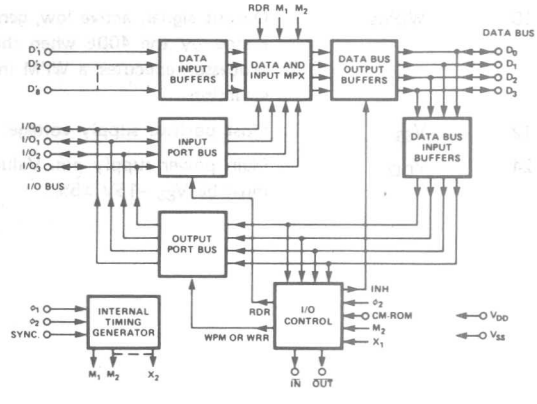
PIN CONFIGURATIONS



4008 BLOCK DIAGRAM



4009 BLOCK DIAGRAM



Pin Description

4008

Pin No.	Designation/ Type of Logic	Description of Function
1-4	D_0-D_3/Neg.	Bidirectional data bus. All address, instruction and data communication between processor and the PROGRAM MEMORY or I/O ports is transmitted on these 4 pins.
7-8	ϕ_1-ϕ_2/Neg.	Non-overlapping clock signals which are used to generate the basic chip timing.
6	SYNC/Neg.	Synchronization input signal driven by SYNC output of processor.
5	CM-ROM/Neg.	Command input driven by CM-ROM output of processor. Used for decoding SRC and I/O instructions.
23-16	A_0-A_7/Pos.	Address output buffers. The demultiplexed address values generated by the 4289 from the address data supplied by the processor at A_1 and A_2.
15-13, 11	C_0-C_3/Pos.	Chip select output buffers. The address data generated by the processor at A_3, or during an SRC are transferred here.
9	F/L/Neg.	Output signal generated by the 4008 to indicate which half-byte of PROGRAM MEMORY is to be operated on.
10	W/Pos.	Output signal, active low, generated by the 4008 when the processor executes a WPM instruction.
12	V_{SS}	Most positive supply voltage.
24	V_{DD}	Main power supply pin. Value must be V_{SS} -15V ±5%.

4009

Pin No.	Designation/ Type of Logic	Description of Function
23-20	D_0-D_3/Neg.	Bidirectional data bus. All address, instruction and data communication between processor and the PROGRAM MEMORY or I/O ports is transmitted on these 4 pins.
5-8, 1-4	D'_1-D'_8/Pos.	The eight bits of instruction from the program memory are transferred on these 4009 pins (most significant bit is D'_8).
14-13	ϕ_1-ϕ_2/Neg.	Non-overlapping clock signals which are used to generate the basic chip timing.
11	SYNC/Neg.	Synchronization input signal driven by SYNC output of processor.
15	CM-ROM/Neg.	Command input driven by CM-ROM output of Processor.
9	IN/Neg.	Output signal, active low, generated by the 4289 when the processor executes an RDR instruction.
10	OUT/Neg.	Output signal, active low (V_{DD}), generated by the 4009 when the processor executes a WRR instruction.
19-16	I/O_0-I/O_3/Pos.	Bidirectional I/O data bus. Data to and from I/O ports or data to write PROGRAM MEMORY are transferred via these pins.
23	V_{DD}	Main power supply pin. Value must be V_{SS} -15V ±5%.
12	V_{SS}	Most positive supply voltage.

Functional Description

The 4008 is the address latch chip which interfaces the 4004 or 4040 to standard PROMs, ROMs and RAMs used for program memory. The 4008 latches the low order eight bits of the program address sent out by the CPU during A1 and A2 time. During A3 time it latches the high order four bits of the program address from the CPU. The low-order eight bits of the program address are then presented at pins A0 through A7 and the high-order four bit (also referred to as page number) are presented at pins C0 through C3. These four bits must be decoded externally and one page of program memory is selected.

The 4009 then transfers the eight bit instruction from program memory to the CPU four bits at a time at M1 and M2. The command signal sent by the CPU activates the 4009 and initiates this transfer.

When the CPU executes an SRC (Send Register Control) instruction, the 4008 responds by storing the I/O address in its eight bit SRC register. The content of this SRC register is always transferred to the address lines (A0 through A7) and the chip select lines (C0 through C3) at X1 time. The appropriate I/O port is then selected by decoding the chip select lines. The IN and OUT lines of the 4009 indicate whether an input or output operation will occur.

The 4009 is primarily an instruction and I/O transfer device. When the CPU executes an RDR (Read ROM Port) instruction, the 4009 will send an input strobe (pin 9) to enable the selected input port. It also enables I/O input buffers to transfer the input data from the I/O bus to the data bus. When the 4009 interprets a WRR (Write ROM Port) instruction, it transfers output data from the CPU to the I/O bus and sends an output strobe (pin 10) to enable the selected output port.

The WPM (Write Program Memory) instruction is used in conjunction with the 4008/4009 to write data into the RAM program memory. When an instruction is to be stored in RAM program memory, it is written in two four-bit segments. The F/L signal from the 4008 keeps track of which half is being written. When the CPU executes a WPM instruction, the chip select lines of the 4008 are jammed with "1111". In the system design this should be designated as the RAM channel. The W line on the 4008 is also activated by the WPM instruction. The previously selected SRC address on line A0 through A7 of the 4008 becomes the address of the RAM word being written. By appropriately decoding the chip select lines, the W line, and F/L, the write strobes can be generated for the memory.

The F/L line is initially high (V_{SS}) when power comes on. It then pulses low (V_{DD}) when every second WPM is executed. A high (V_{SS}) on the F/L lines means that the first four bits (OPR) are being written, and a low means that the last four bits (OPA) are being written. The 4009 transfers the segment of the instruction to the I/O bus at X2 of the WPM instruction. The SRC address sent to RAM is only 8 bits. When more than one page of RAM (256 bytes) is being written, an output port must be used to supply additional address lines for higher order addresses.

Absolute Maximum Ratings*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	-55°C to + 125°C
Input Voltages and Supply Voltage with respect to Vss	+0.5V to -20V
Power Dissipation	1.0 Watt

COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec; $t_{\phi D2}$ = 150 nsec; Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless Otherwise Specified.

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
SUPPLY CURRENT						
I_{DD}	Average Supply Current (4008 only)		10	20	mA	T_A = 25°C
I_{DD}	Average Supply Current (4009 only)		13	30	mA	T_A = 25°C
INPUT CHARACTERISTICS—ALL INPUTS EXCEPT I/O PINS						
I_{LI}	Input Leakage Current			10	μA	V_{IL} = V_{DD}
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS—ALL OUTPUTS EXCEPT I/O PINS						
I_{LO}	Data Bus Output Leakage Current			10	μA	V_{OUT} = -12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitance Load
I_{OL}	Data Lines Sinking Current	8	15		mA	V_{OUT} = V_{SS}
I_{OL}[1]	Address Line Sinking Current (4008 only)	7	13		mA	V_{OUT} = V_{SS}
I_{OL}	In, Out, F/L, Chip Select	1.6	4		mA	V_{OUT} = V_{SS} -4.85
I_{OL}[2]	W Output, Sinking Current (4008 only)	2.5	5		mA	V_{OUT} = V_{SS}
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL} = 0.5mA
R_{OH}	Output Resistance, Data Line "0" Level (4008 only)		150	250	Ω	V_{OUT} = V_{SS} -.5V
R_{OH}	Address, Chip Select Output Resistance, "0" Level (4008 only)		.6	1.2	kΩ	V_{OUT} = V_{SS} -.5V
R_{OH}	Output Resistance, Data Line "0" Level (4009 only)		130	250	Ω	V_{OUT} = V_{SS} -2V
I_{CF}[3]	Address, C/S Output "1" Clamp Current (4008 only)			16	mA	V_{OUT} = V_{SS} -6V
I_{CF}[3]	In, Out "1" Clamp Current (4009 only)			16	mA	V_{OUT} = V_{SS} -6V
I/O INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	
V_{IH}[4]	Input High Voltage	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (4009 only)	V_{DD}		V_{SS}-4.2	V	
I/O OUTPUT CHARACTERISTICS						
V_{OH}	Output High Voltage	V_{SS}-.5V			V	I_{OUT} = 0
R_{OH}	I/O Output "0" Resistance (4009 only)		.25	1.0	kΩ	V_{OUT} = V_{SS} -.5
I_{OL}	I/O Output "1" Sink Current (4009 only)	5	12		mA	V_{OUT} = V_{SS} -.5V
I_{OL}	I/O Output "1" Sink Current (4009 only)	1.6	4		mA	V_{OUT} = V_{SS} -4.85V
I_{CF}	I/O Output "1" Clamp Current (4009 only)			16	mA	V_{OUT} = V_{SS} -6V
CAPACITANCE						
C_{ϕ}	Clock Capacitance		8	15	pF	V_{IN} = V_{SS}
C_{DB}	Data Bus Capacitance		7	10	pF	V_{IN} = V_{SS}
C_{IN}	Input Capacitance (4008 only)			10	pF	V_{IN} = V_{SS}
C_{IN}	Input Capacitance (4009 only)			15	pF	V_{IN} = V_{SS}
C_{OUT}	Output Capacitance			10	pF	V_{IN} = V_{SS}

Notes: 1. The address lines will drive a TTL load if a 470Ω resistor is connected in series between the address output and the TTL input.
2. A 6.8kΩ resistor must be connected between Pin W and V_{DD} for TTL capability.
3. Resistors in series with TTL inputs may be required to limit current into V_{DD} or V_{SS} from TTL input clamp diodes.
4. TTL V_{OH} = 2.4V will ensure 4009 V_{IH} = V_{SS} -1.5 via the 4009 latch. Refer to Figure 3.

A.C. Characteristics

$T_A = 0°C$ to $70°C$, $V_{SS} - V_{DD} = 15V \pm 5\%$

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Test Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t\phi_R$	Clock Rise Time			50	ns	
$t\phi_F$	Clock Fall Times			50	ns	
$t\phi_{PW}$	Clock Width	380		480	ns	
$t\phi_{D1}$	Clock Delay $\phi 1$ to ϕ_2	400		500	ns	
$t\phi_{D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500 pF Data Lines
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT} = 20pF
t_{A1}	Address to Output Delay at A_1, X_1 (4008)			580	ns	C_L = 250pF
t_{A2}	Address to Output Delay A_2 (4008)			580	ns	C_L = 250pF
t_{CS}	Chip Select Output Delay at A_3 (4008)			300	ns	C_L = 50pF
t_{WD}	W Output Delay (4008)			600	ns	C_L = 100pF
t_{FD}	F/L Output Delay (4008)	0.1		1	μs	C_L = 100pF
t_{WI}	Data In Write Time (4009)	470			ns	C_L = 200pF on data bus
t_D	I/O Output Delay (4009)			1.0	μs	C_L = 300pF
t_{S1}	IN Strobe Delay (4009)			450	ns	C_L = 50pF
t_{S2}	OUT Strobe Delay (4009)			1.0	μs	C_L = 50pF

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.

2. t_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.

3. All MCS-40 components which may transmit instruction or data to 4004/4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.

Timing Diagram

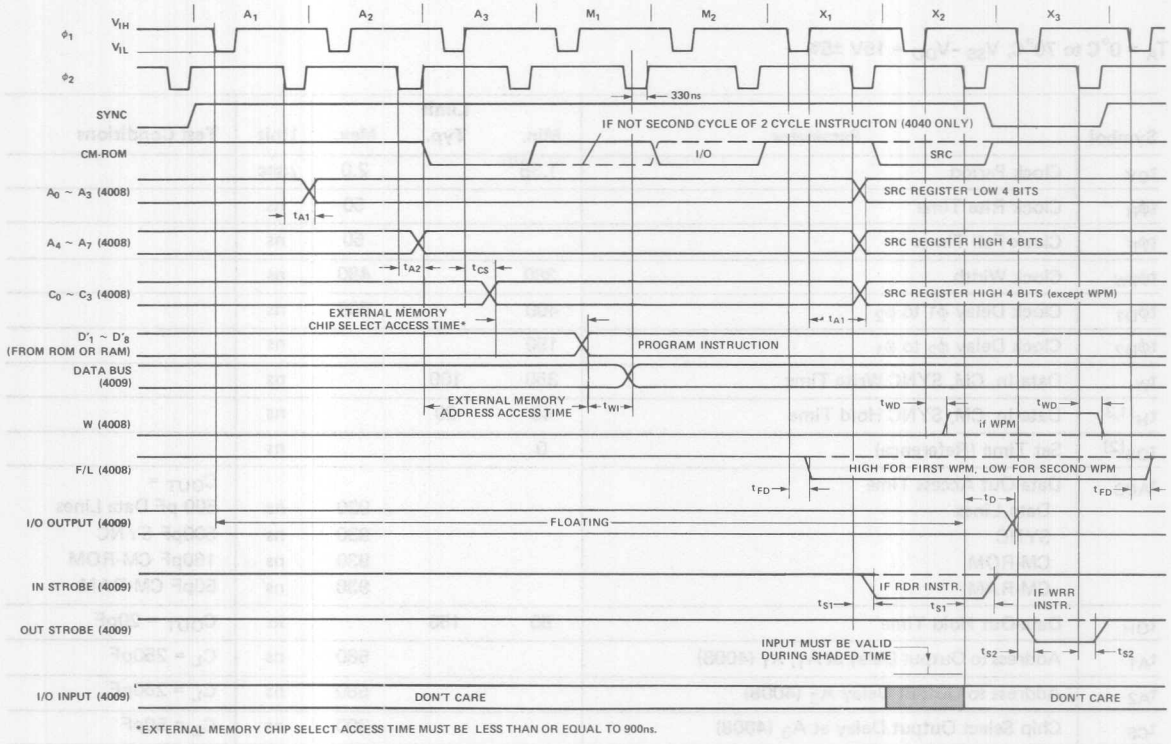


Figure 1. 4008 and 4009 Timing Diagram.

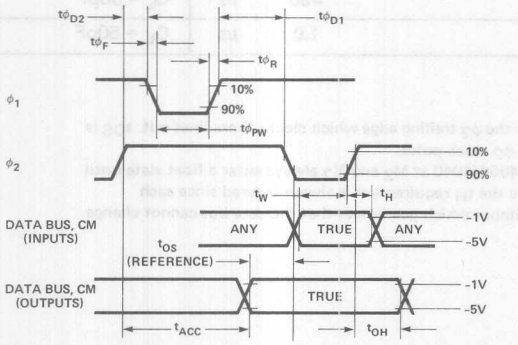
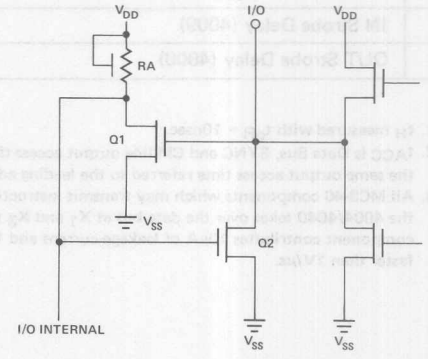


Figure 2. MCS-40 Timing Detail.



EXPLANATION:

WITH V_{SS} = +5V and V_{DD} = -10V, AN EXTERNAL TTL INPUTTING TO THE 4009 ON THE I/O LINE, RAISES THE I/O LINE TO 2.4V. THE Q1-RA INVERTER TURNS "OFF" AND Q2 PULLS THE I/O LINE TO V_{SS}. A LOW TTL SIGNAL OVERRIDES Q2. IF THE TTL OUTPUT GOES TO THE THIRD STATE, THE EXTERNAL I/O LINES REMAIN HIGH THROUGH Q2.

THE PURPOSE OF THIS CIRCUIT IS TO REMOVE RESISTORS TO $V_{CC} = V_{SS}$ ON TTL OUTPUTS, AS R_1 DOES ON 4001/4308 INPUT PORTS.

Figure 3. 4009 I/O Latch.

STANDARD MEMORY INTERFACE

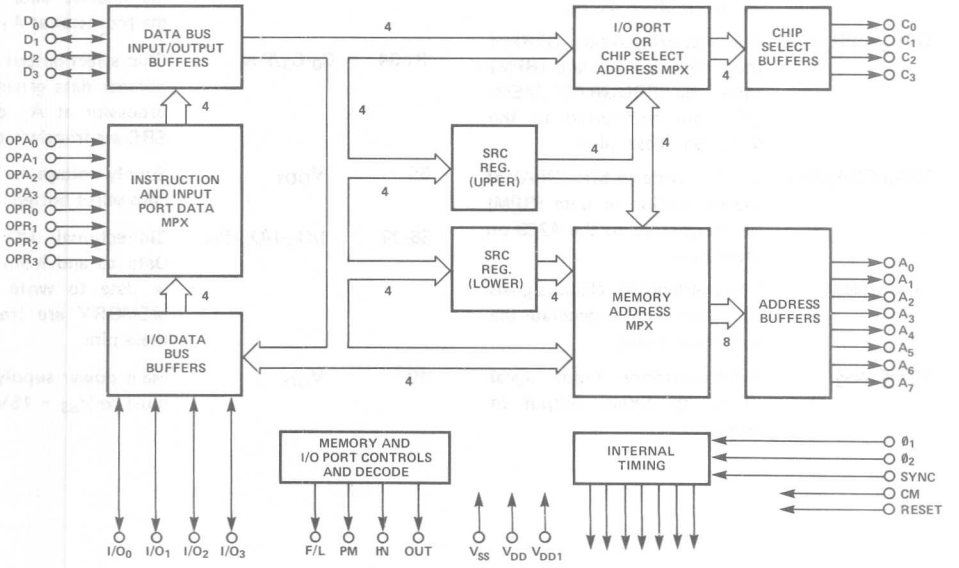
- **Direct Interface to all Standard Memories**
- **Allows Read and Write Program Memory**
- **Single Package Equivalent of 4008/4009**
- **TTL Compatible Address, Chip Select, Program Memory Data Lines**
- **40 Pin Dual In-Line Package**
- **Standard Operating Temperature Range of 0° to 70° C**
- **Also Available With -40° to +85° C Operating Range**

The 4289 standard memory interface and I/O interface enables the CPU devices to utilize standard memory components as program data memory. Notably, PROMs (4702A), RAMs (2102) and ROMs can be arranged in a memory array to facilitate system development. Programs generated using the 4289 interface can be committed to MCS-40™ ROMs (4308 and 4001) with no change to software.

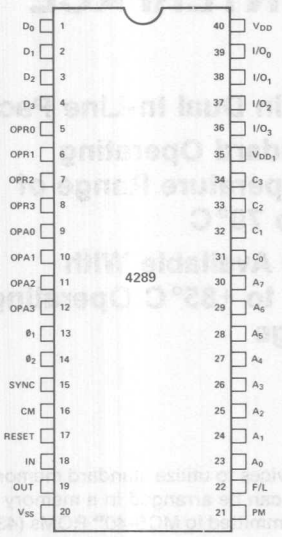
The 4289 also contains a 4 bit bi-directional I/O port and necessary steering logic to multiplex a host of I/O sources to the CPU. The Read and Write Program Memory instruction allows the user to store data and modify program memory. The device directly addresses 4K of program memory. The address is obtained sequentially during A_1-A_3 states of an instruction cycle. The eight bit instruction is presented to the CPU during M_1 and M_2 states of the instruction cycle via the four bit data bus.

The 4289 stores the SRC instruction operand as an I/O address and responds to the ROM I/O instructions (WRR and RDR) by reading or writing data to and from the processor and 4289 I/O bus.

BLOCK DIAGRAM



PIN CONFIGURATION



Pin Description

Pin No.	Designation/ Type of Logic	Description of Function
1-4	D_0-D_3/Neg.	Bidirectional data bus. All address, instruction and data communication between processor and the PROGRAM MEMORY or I/O ports is transmitted on these 4 pins.
5-8	OPR_0-OPR_3/Pos.	The high order 4 bits (OPR) of the instruction or data (RPM) from the PROGRAM MEMORY are transferred to the 4289 on these pins.
9-12	OPA_0-OPA_3/Pos.	The low order 4 bits (OPA) of the instruction or data (RPM) are transferred to the 4289 on these pins.
13-14	ϕ_1-ϕ_2/Neg.	Non-overlapping clock signals which are used to generate the basic chip timing.
15	SYNC/Neg.	Synchronization input signal driven by SYNC output of processor.
16	CM/Neg.	Command input driven by CM-ROM output of processor. Used for decoding SRC and I/O instructions.
17	RESET/Neg.	RESET input. A negative logic "1" level (V_{DD}) applied to this input resets the FIRST/LAST flip-flop.
18	IN/Neg.	Output signal, active low (V_{DD}), generated by the 4289 when the processor executes an RDR or RPM instruction.
19	OUT/Neg.	Output signal, active low (V_{DD}), generated by the 4289 when the processor executes a WRR or WPM instruction.
20	V_{SS}	Most positive supply voltage.
21	PM/Neg.	Output signal, active low (V_{DD}), generated by the 4289 when the processor executes an RPM or WPM instruction.
22	F/L/Neg.	Output signal generated by the 4289 to indicate which half-byte of PROGRAM MEMORY is to be operated on (V_{DD} = OPR, V_{SS} = OPA).
23-30	A_0-A_7/Pos.	Address output buffers. The demultiplexed address values generated by the 4289 from the address data supplied by the processor at A_1 and A_2.
31-34	C_0-C_3/Pos.	Chip select output buffers. The address data generated by the processor at A_3 or during an SRC are transferred here.
35	V_{DD1}	Supply voltage for address and chip select buffers.
36-39	I/O_3-I/O_0/Pos.	Bidirectional I/O data port. Data to and from I/O devices or data to write PROGRAM MEMORY are transferred via these pins.
40	V_{DD}	Main power supply pin. Value must be V_{SS} - 15V ± 5%.

Functional Description

The 4289 enables the 4 bit CPU chip (4004 or 4040) to interface to standard memory components. This allows construction of prototype or small volume systems using electrically programmable ROMs or RAMs in place of 4001 or 4308 mask programmable ROMs. Since 4001s or 4308s also contain up to 16 mask programmable I/O ports, the 4289 has provisions for directly addressing 16 channels of 4 bit I/O ports. In its role as a Memory and I/O interface device, the 4289 provides three different types of operation, namely:

a. Interface to Program Memory for instruction fetch operations.
b. Interface to Input/Output ports for storing or fetching data using WRR, RDR instruction.
c. Interface to R/W Program Memory for program alteration using WPM, RPM instructions. This feature may also be used for storing or fetching data, thus allowing the use of standard R/W RAM for data storage via the 4289.

These three basic operations will be discussed in detail in the following paragraphs.

Instruction Execution

The contents of the data bus at A_1, A_2, and A_3 are latched by the 4289 and transferred to the address and chip select output buffers. The low order address at A_1 is transferred to A_0-A_3 outputs, the middle order address at A_2 is transferred to A_4-A_7 outputs and the high order address at A_3 is transferred to C_0-C_3 outputs. These 12 output lines provide the necessary address and chip select signals to interface to a 4K x 8 bit Program Memory.

The 8 bit word selected by A_0-A_7 and C_0-C_3 is transferred to the processor via the $OPR_{0\text{-}3}$, $OPA_{0\text{-}3}$ input lines and the data output buffer. The high order bits (OPR) are transferred at M_1 and the low order 4 bits (OPA) are transferred at M_2.

The 4289 has been designed to work equally well with either the 4004 or 4040 processor elements. Since the 4040 is provided with two CM-ROM controls which allow it to directly address up to 8K x 8 bits of Program Memory (4K x 8 bits selected by each CM-ROM control), two 4289s would be required for full memory capability. In this case, one 4289 would be controlled by CM-ROM0 and the other by CM-ROM1. The 4289 which receives CM at A3 would be enabled to transfer data at M_1 and M_2.

It should be noted that the two CM-ROM controls permit the simultaneous use of 4001, 4308, and 4289 in the same system. The ROM's 4001 and 4308 can be mixed and assigned to one CM-ROM control line while a single 4289 can be assigned to the other. However, within one CM-ROM control line, 4289, 4001, and 4308 cannot be mixed, since the 4289 does respond to a full 4K of memory by its design and thus would overlap program memory address with the 4001 or 4308.

I/O Port Operation

When the processor executes an I/O port instruction (WRR or RDR), a previously selected I/O port (via an SRC instruction) is enabled to receive or transmit 4 bits of data. In the case of WRR, the selected output port receives the 4 bit contents of the processor accumulator, and in the case of RDR, the selected input port transmits 4 bits of data to the processor accumulator. The 4 bit value sent out at X_2 time of the SRC instruction is used as the port address. Since the 4289 is capable of addressing 16 4 bit I/O ports, it must therefore be capable of storing the SRC address sent by the processor and presenting that address to the external I/O port selection logic for WRR or RDR instructions which follow. To accomplish this, the 4289 behaves as follows:

a. When the processor executes an SRC instruction, the 4289 stores the address sent out by the processor at X_2 and X_3. The contents of the upper 4-bits of the SRC register are transferred during every X_1 time to the chip select lines and are available for subsequent I/O instructions' port selection.
b. When the processor then executes a WRR instruction, the 4289 latches the data sent out by the processor at X_2 and transfers this data to the I/O output buffer. This buffer is enabled during X_3 and transmits the data to the selected output port. So that external port logic may be enabled to receive the data, the 4289 generates the OUT strobe signal.
c. When the processor executes an RDR instruction, the 4289 generates the IN strobe. This enables the selected input port to transmit its data to the I/O bus, where it is latched by the 4289 and transferred to the processor at X_2.

Note that in a system using ROMs, the 4 bit port number is decoded by the ROM chip itself. Where a 4289 is used, the 4 bit port number outputted at the chip select lines C_0-C_3 must be externally decoded to select the appropriate I/O device.

Read/Write Program Memory Operations

If the 4289 is used in conjunction with the 4040, both the WRITE and READ PROGRAM MEMORY (WPM/RPM) functions are directly available (only the WPM is available for 4004 systems). To accomplish these operations, the following are required:

a. A program memory address.
b. The proper control signals.
c. A means of transmitting the data to be stored or fetched.

The 4289 provides all of these as described below.

Program Memory Address

The address for an RPM or WPM operation is provided by the 8 bit contents of the SRC register. Note that the RPM or WPM instruction must have been preceded by an SRC instruction which loaded an 8-bit address into the 4289's SRC register. This 8-bit address is the full address of an 8-bit word in one Read/Write Program Memory page (256 bytes). If more than one page of Read/Write Program Memory is desired, these pages must be selected by external logic controlled via other output ports of the system. At X_1 of every instruction cycle the 8 bit value contained in the SRC register is transferred to the address output buffers A_0-A_7. This address will select 1 out of 256 program memory words.

During execution of WPM or RPM, the 4289 does not transfer the high order 4 bits of the SRC register to C_0-C_3.

Instead, it forces all 4 chip select output buffers to a logic "1" state (positive true logic or V_{SS}). This forcing of C_0-C_3 to all "1s" can be used to indicate the execution of a WPM or RPM instruction. The PM output signal is also generated whenever a proper memory operation (WPM or RPM) is being performed. If only one page of R/W memory is required, the 1111 condition on C_0-C_3 or the PM signal can be used to enable that page. If more than one page is required, an additional output port of the system along with external logic will be necessary to provide the 1 out of 16 page select function.

Since the program memory is organized as 8 bit words, and since RPM and WPM are transmitting only 4 bit words, it is also necessary to specify either the upper or lower half-byte of program memory.

This is done automatically by a FIRST/LAST flip-flop and output signal in the 4289. The state of this flip-flop is used to generate the control signal F/L which determines the proper half-byte of program memory. If F/L is a logic "1" state (V_{DD}), OPR is selected. When F/L is a logic "0" (V_{SS}), OPA is selected. The user can directly reset the FIRST/LAST flip-flop to logic "0" (V_{SS}) in the 4289 by applying a RESET signal.

Starting from a "reset" condition the FIRST/LAST flip-flop automatically toggles after executing either an RPM or WPM instruction. Hence, odd numbered program memory operations select OPA and even numbered program memory operations select OPR (starting with #1 from reset). Alternate WPM and RPM instructions should be used with care since this can cause an out of sequence with the F/L line.

The OUT strobe signal is generated only during WRR and WPM instructions. Hence, the combination of the PM signal (or C_0-C_3 = 1111) and the OUT signal can be used as a WRITE ENABLE for R/W program memory.

Program Memory Data Paths

When the processor executes the WPM instruction, the 4289 latches the data sent out at X_2 by the processor and transfers it via the I/O output buffers to the I/O port. The I/O port must be connected to the data input pins of the R/W memory chips. (Refer to Figure 2 which follows.)

If the processor (4040) executes the RPM instruction, then the entire 8 bit program memory word is transferred to the OPR_0-OPR_3 and OPA_0-OPA_3 inputs of the 4289. Depending on the state of the F/L signal, either the OPA or the OPR half-byte is automatically selected by the 4289.

Data Storage

If Read/Write Memory is interfaced to a 4289 and is used for data storage only, the data is accessed via the WPM and RPM instructions just as Read/Write Program Memory would be accessed. The only difference that the chip select lines C_0-C_3 are never used to select the Read/Write Memory in an instruction fetch operation. The PM pulse would be used to select the Read/Write data memory.

Note that the RAM instructions RDM, WRM, WR0-WR3, RD0-RD3, SBM and ADM cannot be used to access this type of data Read/Write memory.

4008/4009 and 4289 Differences

The functional differences between a 4289 and a 4008/4009 Standard Memory Interface component pair are as follows:

1. The PM pulse of the 4289 (negative logic) is inverted in comparison with the W pulse of the 4008 (positive logic).
2. The W pulse of the 4008 begins in X2 and ends in X3. The 4289's PM pulse begins in X1 and ends in A1.
3. The OUT strobe of the 4289 goes to logical 1 (V_{DD}) for the WRR instructions and the WPM instructions. The OUT strobe of the 4009 goes to logical 1 (V_{DD}) for the WRR instruction only.

4289 Applications

The 4289 can be used to form systems of widely varying complexity. Simple systems containing only one page (256 x 8) of PROGRAM MEMORY and few I/O ports, or more complex systems requiring as many as 32 pages (8K x 8) of memory and 32 I/O ports can readily be implemented. Several examples will be described here.

1. Basic PROM Microcomputer System (Figure 1). This system contains:
   a. 1K x 8 bits of PROGRAM MEMORY (4702A PROM)
   b. 1280 bits of DATA MEMORY (4002 RAM) organized as 16 20-character registers
   c. 4 RAM output ports (4002)
   d. 4 I/O ports.

This system uses a 3205 1 out of 8 decoder to decode the input port addressed by the CPU. Two chip select signals (C_0 and C_1) are combined with the IN signal, which is activated low to indicate an input operation, to select one of four input ports. The 3205 enables one DM 7098 three-state buffer.

In a similar manner, one 3205 and the OUT signal, which is activated to indicate an output operation, are used to select one of four output ports.

2. Standard PROM and RAM Memory System (Figure 2). This system again contains 4 pages of PROM storage but, in addition, has one page of RAM storage which can be used for either PROGRAM or DATA storage by using the WPM/RPM instructions. (The RPM instruction is valid only with the 4040.) The RAM storage has been implemented with two 2101's (256 x 4 static RAM). Notice that separate WRITE ENABLE signals must be generated for the upper and lower half-bytes of RAM.

Note that the inputs to the 2101 RAMs are connected to the 4289 I/O port while their outputs are connected directly to the OPR-OPA lines.

The 2101 RAMs can be chip selected through their active low chip select lines in either of two cases:

1. By an address decode of 4 when the RAMs are addressed as Program Memory for instruction fetch.
2. By the PM signal when addressed as a RAM read or write via an RPM or WPM instruction. For write operations, the TTL logic shown selects one of the

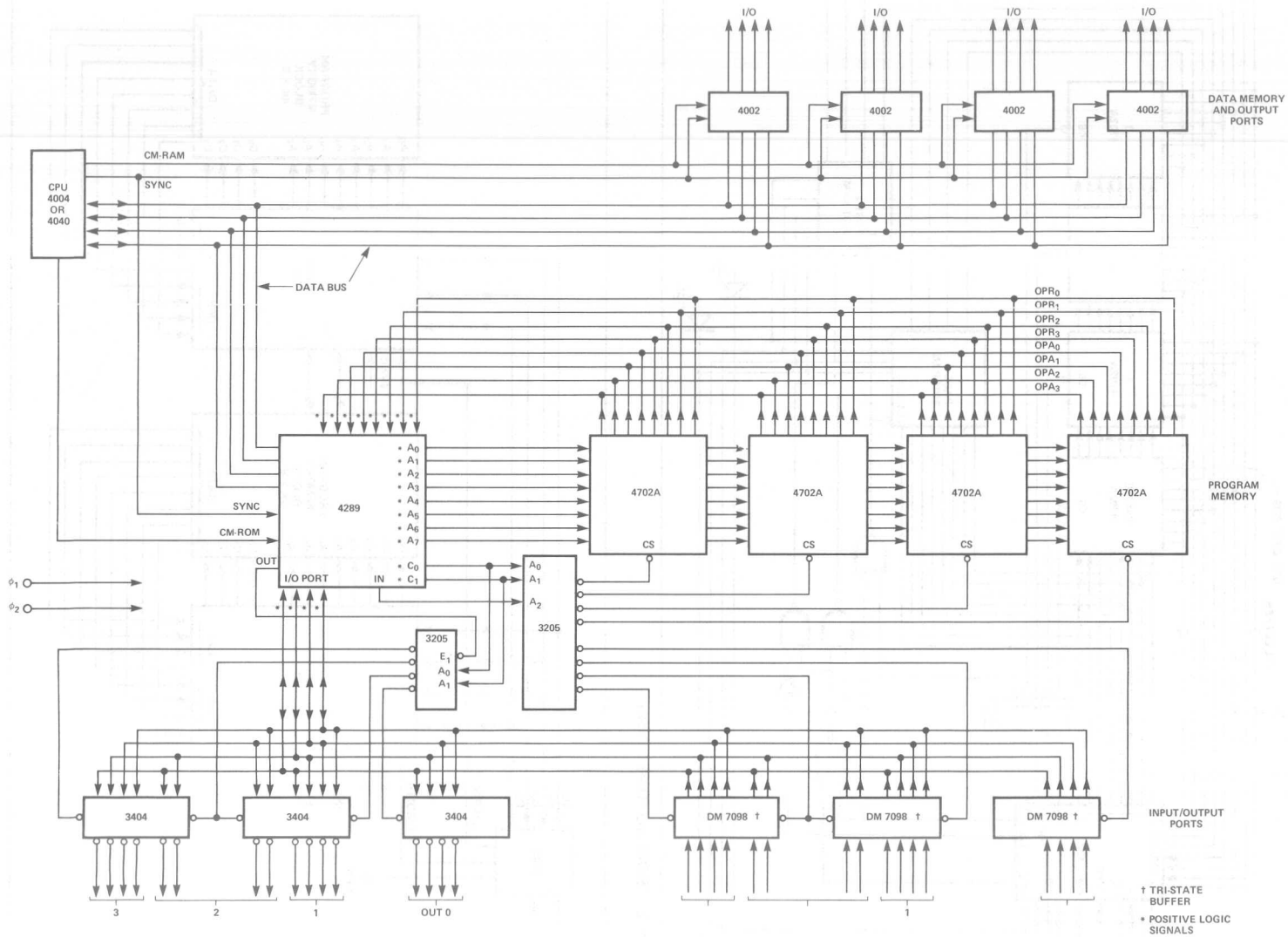


Figure 1. Basic PROM Memory System

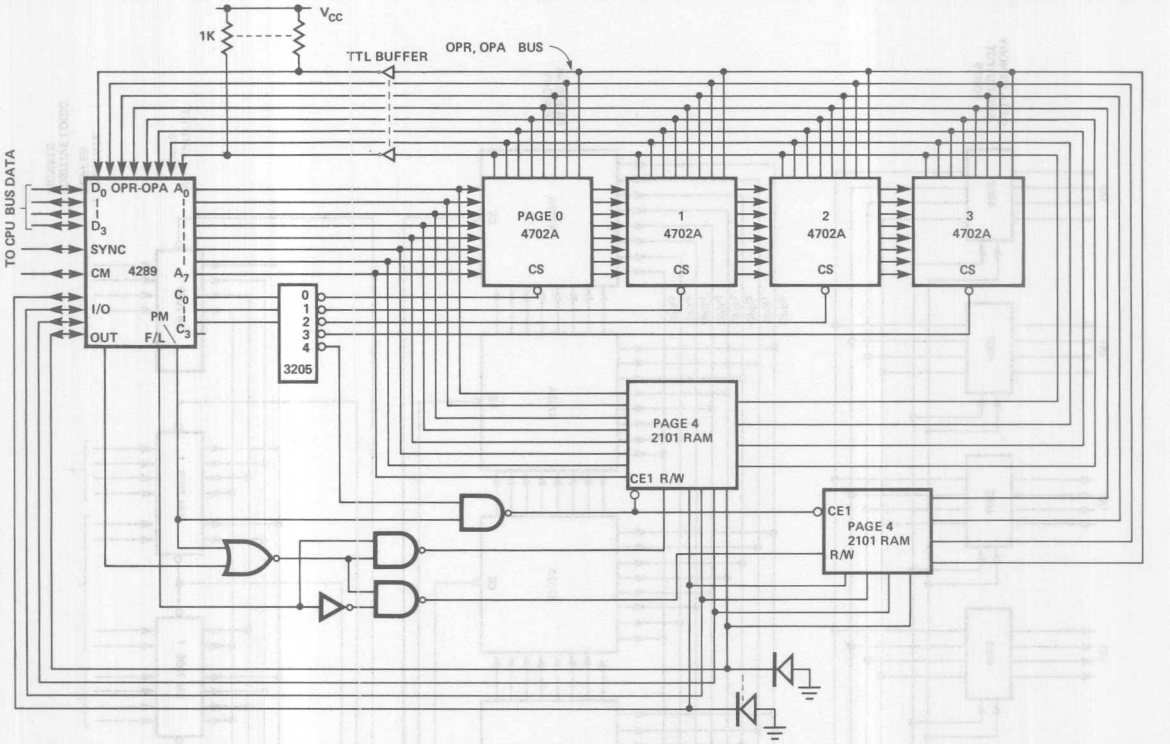


Figure 2. PROM and RAM System.

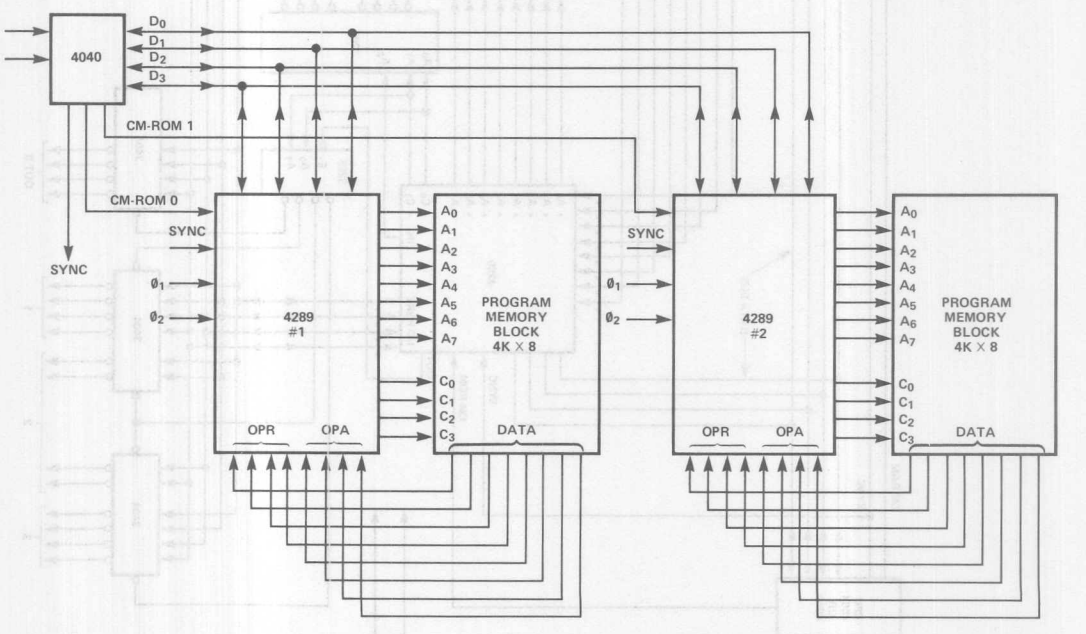


Figure 3. Two Memory Bank System.

two 2101 Read/Write lines according to the F/L signal of the 4289.

The TTL buffers are placed on the data bus to facilitate the compatability between the NMOS RAMs and the PMOS PROMS. The inverters limit the negative excursion of the PROM outputs which may damage the RAMs. The TTL pull-up is required to ensure the V_{IH} threshold level.

3. Two Memory Bank System (Figure 3). Two 4289s are used in this 4040 system giving addressability to a full 8K bytes of PROM memory. In this case each 4289 is controlled from a separate CM-ROM control signal. The CM-ROM$_0$ and CM-ROM$_1$ lines are generated by the 4040. This system cannot be implemented with the 4004.

4289, 4702A System Considerations

1. When utilizing the 4289 with more than six 4702As,a TTL buffer as shown in Figure 4 should be inserted in series with the OPR, OPA lines to achieve maximum clock rate. The buffer may be inverting or non-inverting.

   However, use of a 5.1KΩ resistor on the 4702A output to V_{SS} will allow up to 6 x 4702As to be used without TTL buffers and still achieve maximum clock rate.

2. 4702A access times to meet MCS-40 at $t_{CY} = 1.35\mu$ sec are guaranteed with pure capacitive load of 75pF and with load of 240pF plus a TTL buffer on the 4702A output.

   To operate with more than 6 x 4702A without TTL buffer, the limiting specification is t_{CO} and this increases 5 nsec/pF for capacitance above 75pF; MCS-40 t_{CY} must be increased 2.5ns/pF.

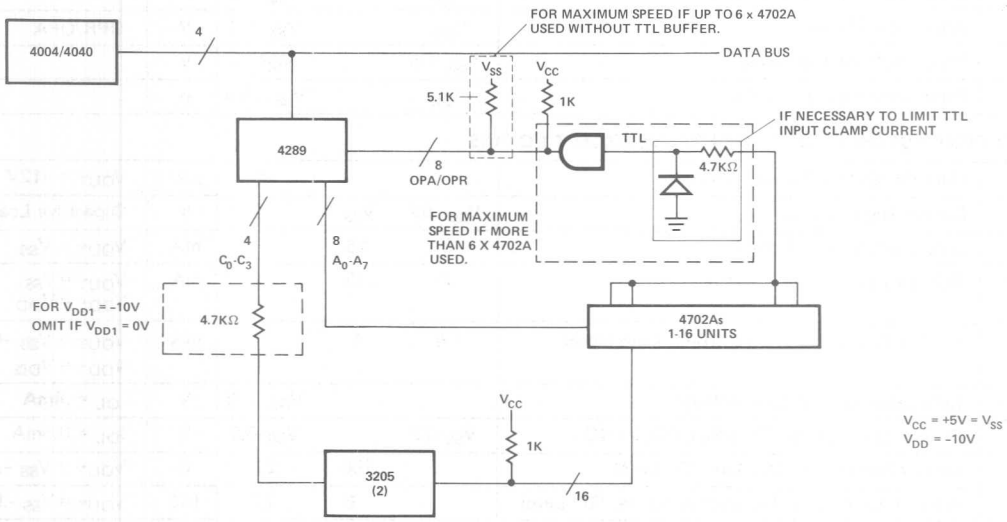


Figure 4. 4289 and 4702A Block Diagram.

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec; $t_{\phi D2}$ = 150 nsec; 4289 V_{DD1} = V_{SS} -5V. Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless Otherwise Specified.

SUPPLY CURRENT

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
I_{DD}	Average Supply Current		30	40	mA	T_A = 25°C
INPUT CHARACTERISTICS—ALL INPUTS EXCEPT I/O PINS						
I_{LI}	Input Leakage Current			10	µA	V_{IL} = V_{DD}
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{ILO}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	OPR/OPA
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS—ALL OUTPUTS EXCEPT I/O PINS						
I_{LO}	Data Bus Output Leakage Current			10	µA	V_{OUT} = -12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitive Load
I_{OL}	Data Lines Sinking Current	8	15		mA	V_{OUT} = V_{SS}
I_{OL}[1]	Address Line Sinking Current	7	13		mA	V_{OUT} = V_{SS}, V_{DD1} = V_{DD}
I_{OL}	In, Out, F/L, PM Sinking Current, Chip Select	1.6	4		mA	V_{OUT} = V_{SS} -4.85 V_{DD1} = V_{DD}
V_{OL}[2]	Chip Select Output Low Voltage			V_{DD1}+.5	V	I_{OL} = .4mA
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL} = 0.5mA
R_{OH}	Output Resistance, Data Line "0" Level		150	250	Ω	V_{OUT} = V_{SS} -.5V
R_{OH}	Address, Chip Select Output Resistance, "0" Level		.6	1.2	kΩ	V_{OUT} = V_{SS} -.5V
I/O INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	µA	
V_{IH}[3]	Input High Voltage	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	
I/O OUTPUT CHARACTERISTICS						
V_{OH}	Output High Voltage	V_{SS}-.5V			V	I_{OUT} = 0
R_{OH}	I/O Output "0" Resistance		.25	1.0	kΩ	V_{OUT} = V_{SS}-.5
I_{OL}	I/O Output "1" Sink Current	5	12		mA	V_{OUT} = V_{SS} -.5
I_{OL}	I/O Output "1" Sink Current	1.6	4		mA	V_{OUT} = V_{SS} -4.85V
I_{CF}	I/O Output "1" Clamp Current			10	mA	V_{OUT} = V_{SS} -6V

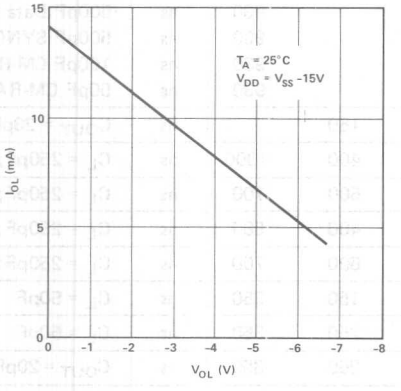
Notes: 1. The address lines will drive a TTL load if a 470Ω resistor is connected in series between the address output and the TTL input.
2. 4289 Address (A_0-A_7) Outputs are also tied to V_{DD1} but are tested with capacitive load only.
3. TTL V_{OH} = 2.4V will ensure 4289 V_{IH} = V_{SS} -1.5V via the 4289 latch. Refer to Figure 5.

D.C. and Operating Characteristics (Continued)

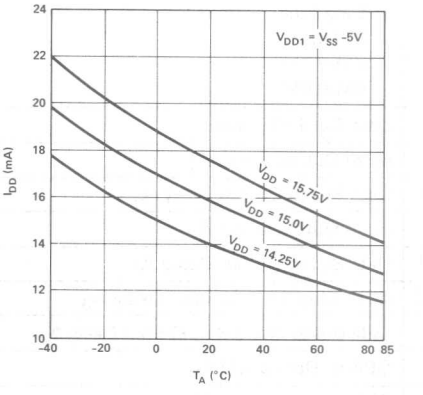
CAPACITANCE

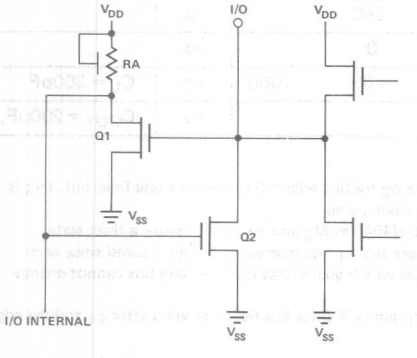
Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
C_ϕ	Clock Capacitance		14	20	pF	$V_{IN} = V_{SS}$
C_{DB}	Data Bus Capacitance		7	10	pF	$V_{IN} = V_{SS}$
C_{IN}	Input Capacitance			15	pF	$V_{IN} = V_{SS}$
C_{OUT}	Output Capacitance			10	pF	$V_{IN} = V_{SS}$

4289 I_{OL} VS. V_{OL} (FOR I/O PINS)



4289 I_{DD} VS. TEMPERATURE





EXPLANATION:

WITH V_{SS} = +5V and V_{DD} = -10V, AN EXTERNAL TTL INPUTTING TO THE 4289 ON THE I/O LINE, RAISES THE I/O LINE TO 2.4V. THE Q1-RA INVERTER TURNS "OFF" AND Q2 PULLS THE I/O LINE TO V_{SS}. A LOW TTL SIGNAL OVERRIDES Q2. IF THE TTL OUTPUT GOES TO THE THIRD STATE, THE EXTERNAL I/O LINES REMAIN HIGH THROUGH Q2.

THE PURPOSE OF THIS CIRCUIT IS TO REMOVE RESISTORS TO V_{CC} = V_{SS} ON TTL OUTPUTS, AS R_I DOES ON 4001/4308 INPUT PORTS.

Figure 5. 4289 I/O Latch.

A.C. Characteristics

T_A = 0°C to 70°C, V_{SS} -V_{DD} = 15V ±5%

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Test Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t\phi_R$	Clock Rise Time			50	ns	
$t\phi_F$	Clock Fall Time			50	ns	
$t\phi_{PW}$	Clock Width	380		480	ns	
$t\phi_{D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t\phi_{D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT} = 20pF
t_{A1}[4]	ϕ_1 to Output Delay A_1		400	1000	ns	C_L = 250pF; A_0-A_3
t_{TA1}[4]	Data Bus to Output Delay A_1		500	700	ns	C_L = 250pF; A_0-A_3
t_{A2}[4]	ϕ_1 to Output Delay A_2		400	580	ns	C_L = 250pF; A_4-A_7
t_{TA2}[4]	Data Bus to Output Delay A_2		500	700	ns	C_L = 250pF; A_4-A_7
t_{CS}[4,5]	ϕ_1 to Chip Select Output Delay A_3		150	350	ns	C_L = 50pF
t_{TC}[4,5]	Data Bus to Chip Select Output Delay A_3		250	350	ns	C_L = 50pF
t_{WID}	OPR to Data Bus Delay		250	350	ns	C_{OUT} = 20pF, Data Bus
t_{SRC}	Output Delay at X_1 Time		400	700	ns	C_L = 250pF
t_{S1}	IN Strobe Delay Time			500	ns	C_L = 50pF
t_{S2}	OUT Strobe Delay Time, Falling			500	ns	C_L = 50pF
t_{FD}	F/L and PM Delay Time		300	500	ns	C_L = 100pF
$t_{W,I/O}$	I/O Input Write Time	400	250		ns	
$t_{H,I/O}$	I/O Input Hold Time	40	0		ns	
$t_{D,I/O}$	I/O Output Delay Time		400	1000	ns	C_L = 300pF
t_{WI}	Data In Write Time	350			ns	C_{OUT} = 200pF, Data Bus

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.

2. T_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.

3. All MCS-40 components which may transmit instruction or data to 4004/4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.

4. t_{A1}, t_{A2}, t_{CS} apply if Data Bus is valid before ϕ_1 trailing edge. t_{TA}, t_{TC} apply if Data Bus becomes valid after ϕ_1 trailing edge.

5. Measured at output of 3205 decoder.

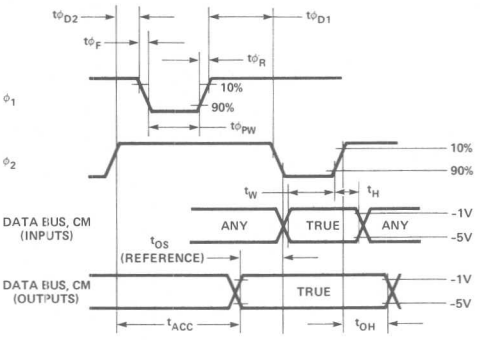


Figure 6. MCS-40 Timing Detail.

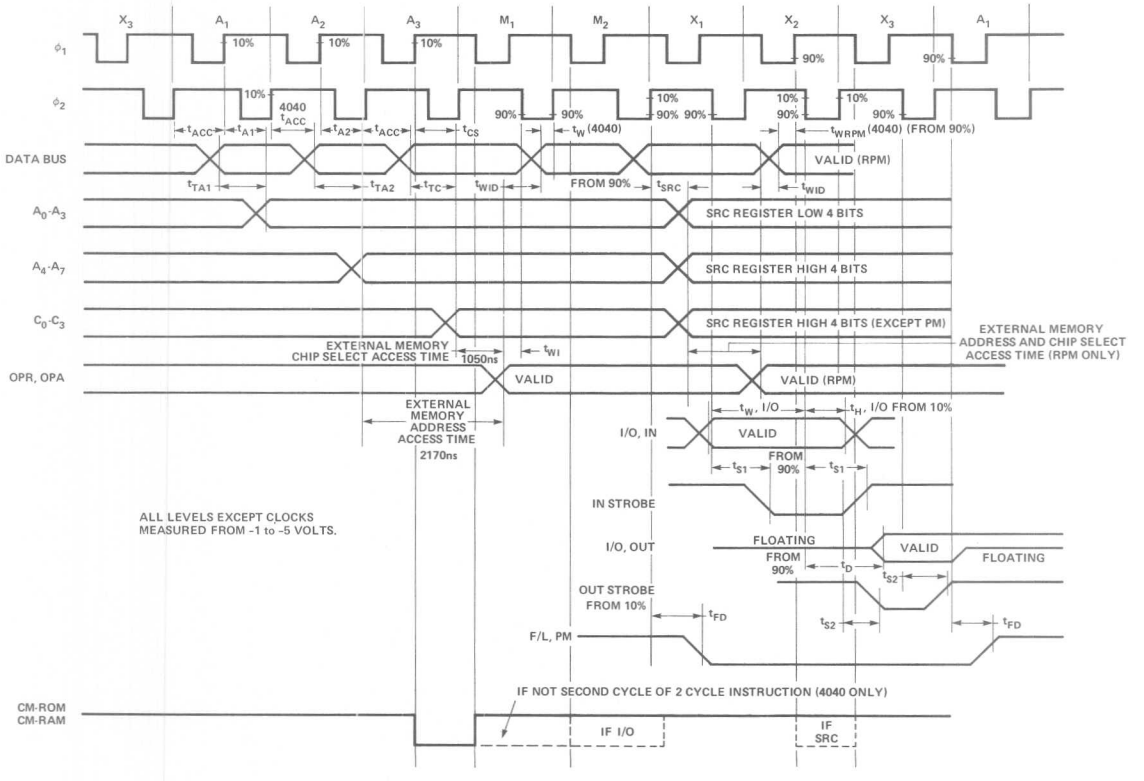
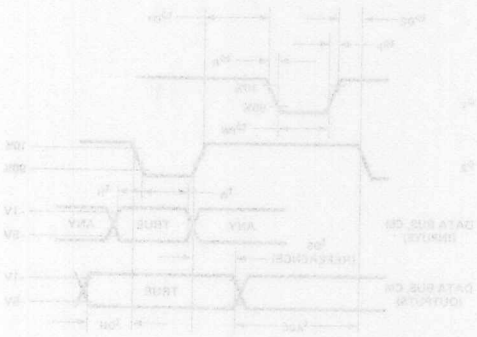
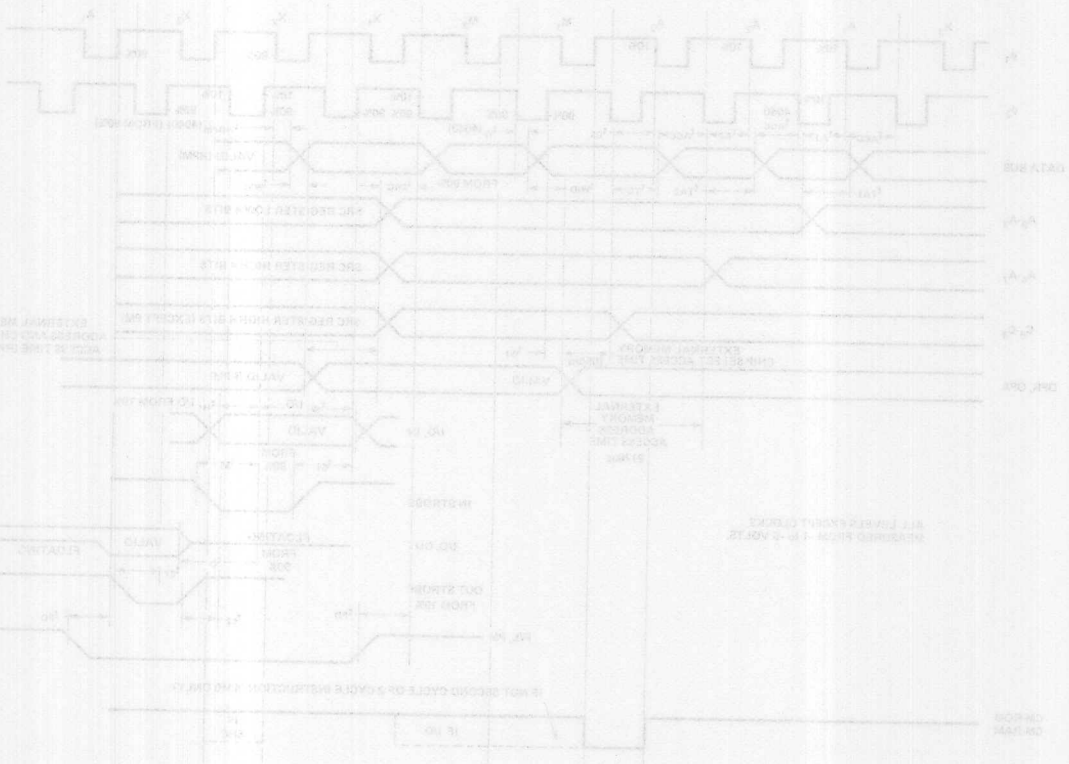


Figure 7. MCS-40 Timing Diagram for 4289.





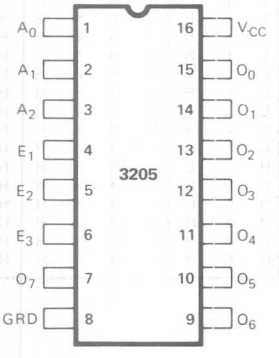
HIGH SPEED 1 OUT OF 8 BINARY DECODER

- **I/O Port or Memory Selector**
- **Simple Expansion — Enable Inputs**
- **High Speed Schottky Bipolar Technology — 18ns Max. Delay**
- **Directly Compatible with TTL Logic Circuits**
- **Low Input Load Current — .25 mA max., 1/6 Standard TTL Input Load**
- **Minimum Line Reflection — Low Voltage Diode Input Clamp**
- **Outputs Sink 10 mA min.**
- **16-Pin Dual-In-Line Ceramic or Plastic Package**

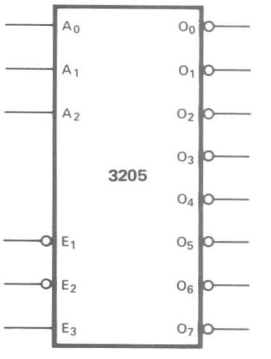
The 3205 decoder can be used for expansion of systems which utilize input ports, output ports, and memory components with active low chip select input. When the 3205 is enabled, one of its eight outputs goes "low", thus a single row of a memory system is selected. The 3 chip enable inputs on the 3205 allow easy system expansion. For very large systems, 3205 decoders can be cascaded such that each decoder can drive eight other decoders for arbitrary memory expansions.

The Intel® 3205 is packaged in a standard 16 pin dual-in-line package; and its performance is specified over the temperature range of 0°C to +75°C, ambient. The use of Schottky barrier diode clamped transistors to obtain fast switching speeds results in higher performance than equivalent devices made with a gold diffusion process.

PIN CONFIGURATION



LOGIC SYMBOL



PIN NAMES

A_0-A_2	ADDRESS INPUTS
$\overline{E_1}$-$\overline{E_3}$	ENABLE INPUTS
$\overline{O_0}$-$\overline{O_7}$	DECODED OUTPUTS

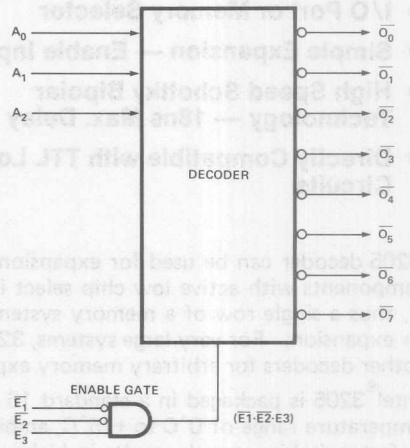
ADDRESS			ENABLE			OUTPUTS							
A_0	A_1	A_2	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

Functional Description

Decoder

The 3205 contains a one out of eight binary decoder. It accepts a three bit binary code and by gating this input, creates an exclusive output that represents the value of the input code.

For example, if a binary code of 101 was present on the A0, A1 and A2 address input lines, and the device was enabled, an active low signal would appear on the $\overline{05}$ output line. Note that all of the other output pins are sitting at a logic high, thus the decoded output is said to be exclusive. The decoders outputs will follow the truth table shown below in the same manner for all other input variations.



Enable Gate

When using a decoder it is often necessary to gate the outputs with timing or enabling signals so that the exclusive output of the decoded value is synchronous with the overall system.

The 3205 has a built-in function for such gating. The three enable inputs ($\overline{E1}$, $\overline{E2}$, E3) are ANDed together and create a single enable signal for the decoder. The combination of both active "high" and active "low" device enable inputs provides the designer with a powerfully flexible gating function to help reduce package count in his system.

ADDRESS			ENABLE			OUTPUTS							
A_0	A_1	A_2	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

Absolute Maximum Ratings*

Temperature Under Bias:	Ceramic	−65°C to +125°C
	Plastic	−65°C to +75°C
Storage Temperature		−65°C to +160°C
All Output or Supply Voltages		−0.5 to +7 Volts
All Input Voltages		−1.0 to +5.5 Volts
Output Currents		125 mA

*COMMENT

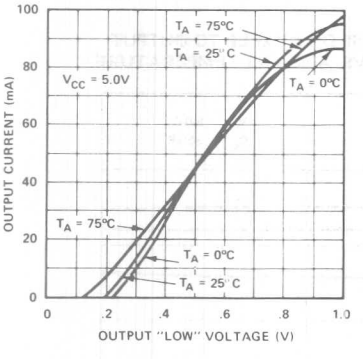
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics T_A = 0°C to +75°C, V_{CC} = 5.0V ±5%

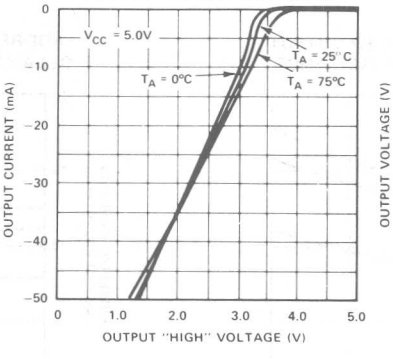
SYMBOL	PARAMETER	LIMIT MIN.	LIMIT MAX.	UNIT	TEST CONDITIONS
I_F	INPUT LOAD CURRENT		−0.25	mA	V_{CC} = 5.25V, V_F = 0.45V
I_R	INPUT LEAKAGE CURRENT		10	µA	V_{CC} = 5.25V, V_R = 5.25V
V_C	INPUT FORWARD CLAMP VOLTAGE		−1.0	V	V_{CC} = 4.75V, I_C = −5.0 mA
V_{OL}	OUTPUT "LOW" VOLTAGE		0.45	V	V_{CC} = 4.75V, I_{OL} = 10.0 mA
V_{OH}	OUTPUT HIGH VOLTAGE	2.4		V	V_{CC} = 4.75V, I_{OH} = −1.5 mA
V_{IL}	INPUT "LOW" VOLTAGE		0.85	V	V_{CC} = 5.0V
V_{IH}	INPUT "HIGH" VOLTAGE	2.0		V	V_{CC} = 5.0V
I_{SC}	OUTPUT HIGH SHORT CIRCUIT CURRENT	−40	−120	mA	V_{CC} = 5.0V, V_{OUT} = 0V
V_{OX}	OUTPUT "LOW" VOLTAGE @ HIGH CURRENT		0.8	V	V_{CC} = 5.0V, I_{OX} = 40 mA
I_{CC}	POWER SUPPLY CURRENT		70	mA	V_{CC} = 5.25V

Typical Characteristics

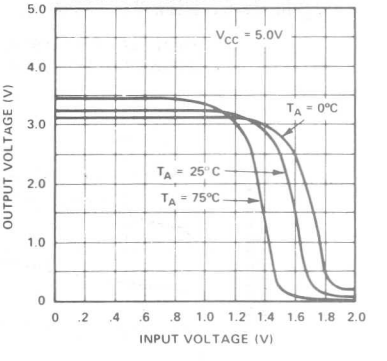
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



DATA TRANSFER FUNCTION



Switching Characteristics

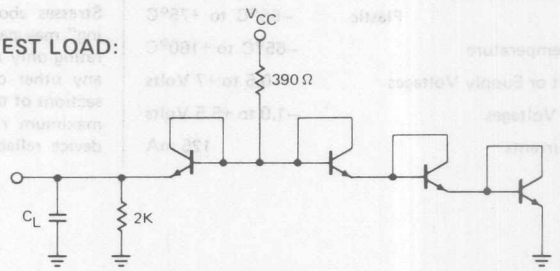
CONDITIONS OF TEST:

Input pulse amplitudes: 2.5V

Input rise and fall times: 5 nsec between 1V and 2V

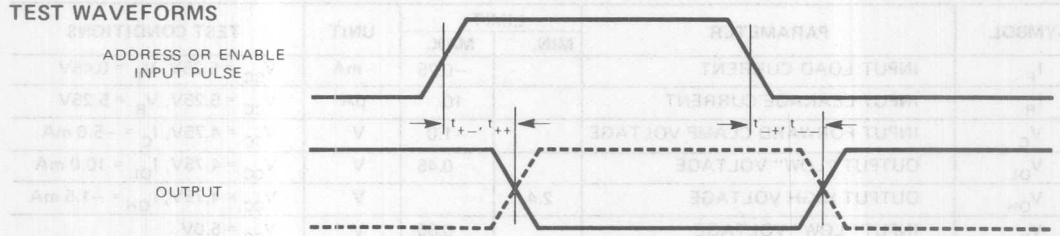
Measurements are made at 1.5V

TEST LOAD:



All Transistors 2N2369 or Equivalent. C_L = 30 pF

TEST WAVEFORMS



A.C. Characteristics

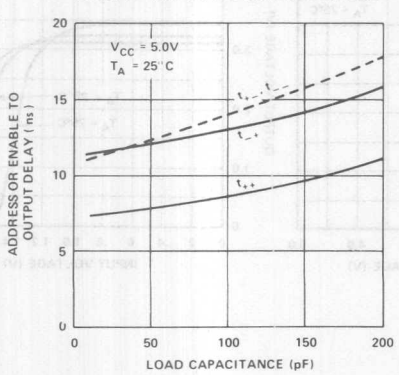
T_A = 0°C to +75°C, V_{CC} = 5.0V ±5% unless otherwise specified.

SYMBOL	PARAMETER		MAX. LIMIT	UNIT	TEST CONDITIONS
t_{++}	ADDRESS OR ENABLE TO OUTPUT DELAY		18	ns	
t_{-+}			18	ns	
t_{+-}			18	ns	
t_{--}			18	ns	
C_{IN} [1]	INPUT CAPACITANCE	P8205	4(typ.)	pF	f = 1 MHz, V_{CC} = 0V
		C8205	5(typ.)	pF	V_{BIAS} = 2.0V, T_A = 25°C

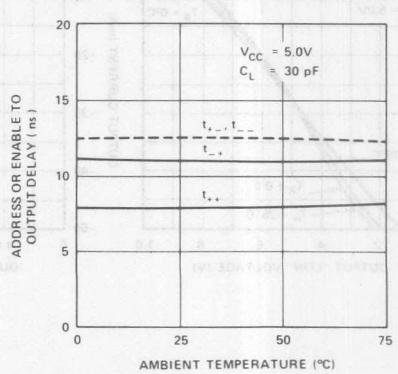
1. This parameter is periodically sampled and is not 100% tested.

Typical Characteristics

ADDRESS OR ENABLE TO OUTPUT DELAY VS. LOAD CAPACITANCE



ADDRESS OR ENABLE TO OUTPUT DELAY VS. AMBIENT TEMPERATURE



intel®

3214

PRIORITY INTERRUPT CONTROL UNIT

- Eight Priority Levels
- Current Status Register
- Priority Comparator
- Fully Expandable
- High Performance (50ns)
- 24-Pin Dual In-Line Package

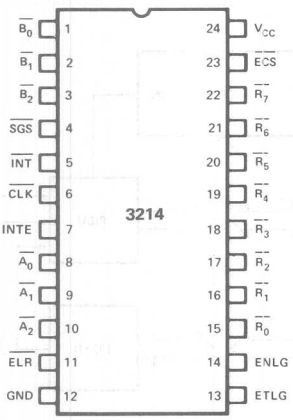
The 3214 is an eight level priority interrupt control unit designed to simplify interrupt driven microcomputer systems.

The PICU can accept eight requesting levels; determine the highest priority, compare this priority to a software controlled current status register and issue an interrupt to the system along with vector information to identify the service routine.

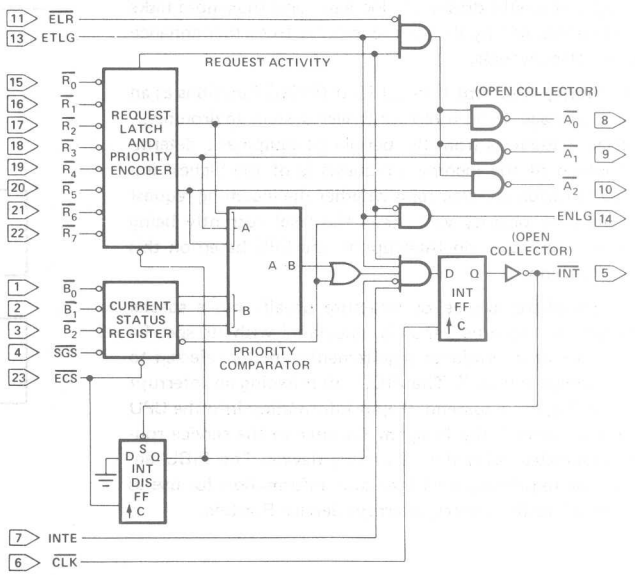
The 3214 is fully expandable by the use of open collector interrupt output and vector information. Control signals are also provided to simplify this function.

The PICU is designed to support a wide variety of vectored interrupt structures and reduce package count in interrupt driven microcomputer systems.

PIN CONFIGURATION



LOGIC DIAGRAM



PIN NAMES

INPUTS:		
$\overline{R_0}$-$\overline{R_7}$	REQUEST LEVELS (R_7 HIGHEST PRIORITY)	
$\overline{B_0}$-$\overline{B_2}$	CURRENT STATUS	
$\overline{SGS}$	STATUS GROUP SELECT	
$\overline{ECS}$	ENABLE CURRENT STATUS	
INTE	INTERRUPT ENABLE	
$\overline{CLK}$	CLOCK (INT F-F)	
$\overline{ELR}$	ENABLE LEVEL READ	
ETLG	ENABLE THIS LEVEL GROUP	
OUTPUTS:		
$\overline{A_0}$-$\overline{A_2}$	REQUEST LEVELS	OPEN COLLECTOR
$\overline{INT}$	INTERRUPT (ACT. LOW)	OPEN COLLECTOR
ENLG	ENABLE NEXT LEVEL GROUP	

INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient method so that large amounts of the total systems tasks can be assumed by the microcomputer with little or no effect on throughput.

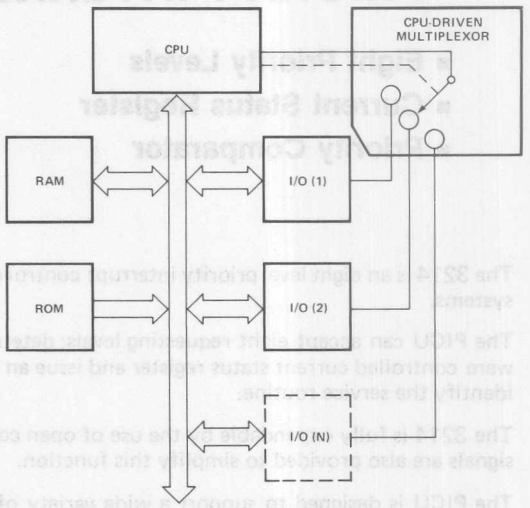
The most common method of servicing such devices is the **Polled** approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, the processor would resume exactly where it left off.

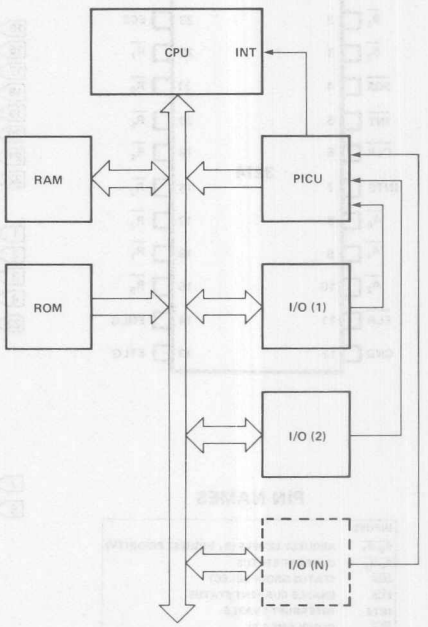
This method is called **Interrupt**. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Priority Interrupt Control Unit (PICU) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced and issues an Interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PICU, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. The PICU encodes the requesting level into such information for use as a "vector" to the correct Interrupt Service Routine.



Polled Method



Interrupt Method

Functional Description

General

The 3214 is a device specifically designed for use in real time, interrupt driven, microcomputer systems. Basically it is an eight (8) level priority control unit that can accept eight different interrupt requests, determine which has the highest priority, compare that level to a software maintained current status register and issue an interrupt to the system based on this comparison along with vector information to indicate the location of the service routine.

Priority Encoder

The eight requests inputs, which are active low, come into the Priority Encoder. This circuit determines which request input is the most important (highest priority) as preassigned by the designer. ($\overline{R7}$) is the highest priority input to the 3214 and ($\overline{R0}$) is the lowest. The logic of the Priority Encoder is such that if two or more input levels arrive at the same time then the input having the highest priority will take precedence and a three bit output corresponding to the active level (modulo 8) will be sent out. The Priority Encoder also contains a latch to store the request input. This latch is controlled by the Interrupt Disable Flip-flop so that once an interrupt has been issued by the 3214 the request latch is no longer open. (Note that the latch does not store inactive requests. In order for a request to be monitored by the 3214 it must remain present until it has been serviced.)

Current Status Register

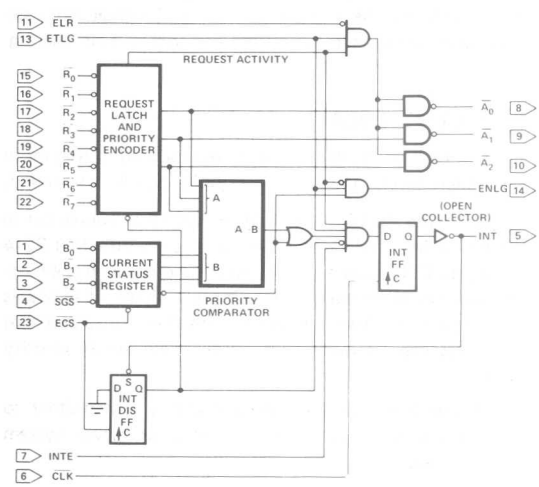
In an interrupt driven microcomputer system it is important to not only prioritize incoming requests but to ascertain whether such a request is a higher priority than the interrupt currently being serviced.

The Current Status Register is a simple 4-bit latch that is treated as an addressable output port by the microcomputer system. It is loaded when the $\overline{ECS}$ input goes low.

Maintenance of the Current Status Register is performed as a portion of the service routine. Basically, when an interrupt is issued to the system the programmer outputs a binary code (modulo 8) that is the complement of the interrupt level. This value is stored in the Current Status Register and is compared to all further prioritized incoming requests by the Priority Comparator. In essence, a copy of the current interrupt level is written into the 3214 to be used as a reference for comparison. There is no restriction to this maintenance. Other level values can be written into this register as references so that groups of interrupt requests may be disallowed under complete control of the programmer.

Note that the fourth bit in the register is $\overline{SGS}$. This input is part of the value written out by the programmer and performs a special function. The Priority Comparator will only issue an output that indicates the request level is greater than the Current Status Register. If both comparator inputs are equal to zero, no output will be present. The $\overline{SGS}$ input allows the programmer to, in effect, disable this comparison and allow the 3214 to issue an interrupt to the system that is based only on the logic of the priority encoder.

LOGIC DIAGRAM



Control Signals

The 3214 also has several inputs that enable the designer to synchronize the interrupt issued to the microprocessor and to allow or disallow such an issuance. Also, signals are provided that permit simple expansion to other 3214s so that more than eight levels can be controlled.

INTE, $\overline{CLK}$

The INTE (Interrupt Enable) input allows the designer to "shutoff" the interrupt system under control of external logic or possibly under software maintenance. A "zero" on this line will not allow interrupts to be issued to the microcomputer system.

The $\overline{CLK}$ (Clock) input is actually the trigger that strobes the Interrupt Flip-Flop. It can be connected to one of the clocks of the microprocessor so that the interrupt issued meets the CPU set-up time specification. Note that due to the gating of the input to the Interrupt Flip-Flop the $\overline{INT}$ output will only be active for the time of a single clock period, so external latching may be required to hold this signal.

$\overline{ELR}$, ETLG, ENGL

These three signals allow 3214s to be cascaded so that more than eight levels of interrupt requests can be controlled.

Basically, the ENLG output of one 3214 is connected to the ETLG input of the next and so on, with the first 3214 having its ETLG input pulled "high" and assigned the highest priority. When the ENLG output is "high" it indicates that there is no interrupt pending on that device and that interrupts can be monitored on the next lower priority 3214.

This "cascading" can be expanded almost indefinitely to accomodate even the largest of interrupt driven system architectures.

$\overline{A0}$, $\overline{A1}$, $\overline{A2}$

In order to identify which device has interrupted the processor so that the service routine associated with it can be addressed, a pointer or "vector" must accompany the interrupt issued to the microcomputer system.

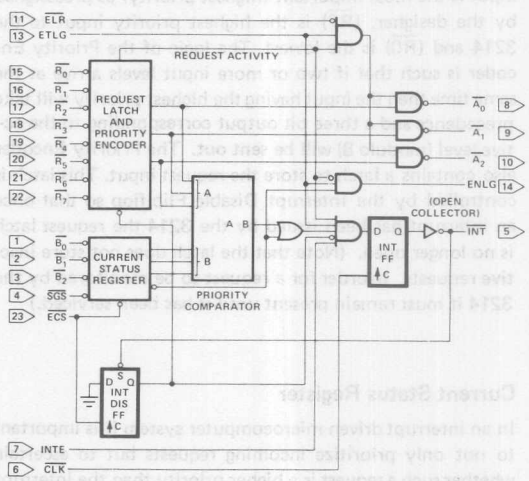
The $\overline{A0}$, $\overline{A1}$ and $\overline{A2}$ outputs represent the complement of the active interrupt level (modulo 8). By using these signals to encode the special instruction, RST, the program counter of the microprocessor, can point to the location of the service routine. Note that these three outputs are gated by the $\overline{ELR}$ input and are open collector so that expansion is simplified.

$\overline{INT}$

The $\overline{INT}$ output of the 3214 is the signal that is issued to the microprocessor to initiate the interrupt sequence. As soon a $\overline{INT}$ is active the INT DIS FF is set, inhibiting further requests from entering the Request Latch. Only the writing out of the current status information by strobing the $\overline{ECS}$ input will clear the INT DIS FF and allow requests to enter the latch.

Note that $\overline{INT}$ is also open collector so that when cascaded to other 3214s an interrupt in any of the active devices will set all INT DIS FFs in the entire array.

LOGIC DIAGRAM



Absolute Maximum Ratings*

Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . -65°C to +150°C
All Output and Supply Voltages . . . . . -0.5V to +7V
All Input Voltages . . . . . -1.0V to +5.5V
Output Currents . . . . . 100 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specifications is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

D.C. and Operating Characteristics T_A = 0°C to +70°C, V_{CC} = 5V ±5%.

Symbol	Parameter		Limits			Unit	Conditions
			Min.	Typ.[1]	Max.		
V_C	Input Clamp Voltage (all inputs)				-1.0	V	I_C=-5mA
I_F	Input Forward Current:	ETLG input all other inputs		-.15 -.08	-0.5 -0.25	mA mA	V_F=0.45V
I_R	Input Reverse Current:	ETLG input all other inputs			80 40	μA μA	V_R=5.25V
V_{IL}	Input LOW Voltage:	all inputs			0.8	V	V_{CC}=5.0V
V_{IH}	Input HIGH Voltage:	all inputs	2.0			V	V_{CC}=5.0V
I_{CC}	Power Supply Current			90	130	mA	See Note 2.
V_{OL}	Output LOW Voltage:	all outputs		.3	.45	V	I_{OL}=15mA
V_{OH}	Output HIGH Voltage:	ENLG output	2.4	3.0		V	I_{OH}=-1mA
I_{OS}	Short Circuit Output Current:	ENLG output	-20	-35	-55	mA	V_{OS}=0V, V_{CC}=5.0V
I_{CEX}	Output Leakage Current:	$\overline{INT}$ and $\overline{A_0}$-$\overline{A_2}$			100	μA	V_{CEX}=5.25V

NOTES:
1. Typical values are for T_A = 25°C, V_{CC} = 5.0V.
2. B_0-B_2, $\overline{SGS}$, CLK, $\overline{R_0}$-$\overline{R_4}$ grounded, all other inputs and all outputs open.

A.C. Characteristics $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Unit
t_{CY}	$\overline{CLK}$ Cycle Time	80	50		ns
t_{PW}	$\overline{CLK}$, $\overline{ECS}$, $\overline{INT}$ Pulse Width	25	15		ns
t_{ISS}	INTE Setup Time to $\overline{CLK}$	16	12		ns
t_{ISH}	INTE Hold Time after $\overline{CLK}$	20	10		ns
t_{ETCS}[2]	ETLG Setup Time to $\overline{CLK}$	25	12		ns
t_{ETCH}[2]	ETLG Hold Time After $\overline{CLK}$	20	10		ns
t_{ECCS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	80	50		ns
t_{ECCH}[3]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{ECRS}[3]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	110	70		ns
t_{ECRH}[3]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			
t_{ECSS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	75	70		ns
t_{ECSH}[2]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{DCS}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{CLK}$	70	50		ns
t_{DCH}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{CLK}$	0			ns
t_{RCS}[3]	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{CLK}$	90	55		ns
t_{RCH}[3]	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{CLK}$	0			ns
t_{ICS}	$\overline{INT}$ Setup Time to $\overline{CLK}$	55	35		ns
t_{CI}	$\overline{CLK}$ to $\overline{INT}$ Propagation Delay		15	25	ns
t_{RIS}[4]	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{INT}$	10	0		ns
t_{RIH}[4]	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{INT}$	35	20		ns
t_{RA}	$\overline{R_0}$-$\overline{R_7}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		80	100	ns
t_{ELA}	$\overline{ELR}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		40	55	ns
t_{ECA}	$\overline{ECS}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		100	120	ns
t_{ETA}	ETLG to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		35	70	ns
t_{DECS}[4]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{ECS}$	15	10		ns
t_{DECH}[4]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{ECS}$	15	10		ns
t_{REN}	$\overline{R_0}$-$\overline{R_7}$ to ENLG Propagation Delay		45	70	ns
t_{ETEN}	ETLG to ENLG Propagation Delay		20	25	ns
t_{ECRN}	$\overline{ECS}$ to ENLG Propagation Delay		85	90	ns
t_{ECSN}	$\overline{ECS}$ to ENLG Propagation Delay		35	55	ns

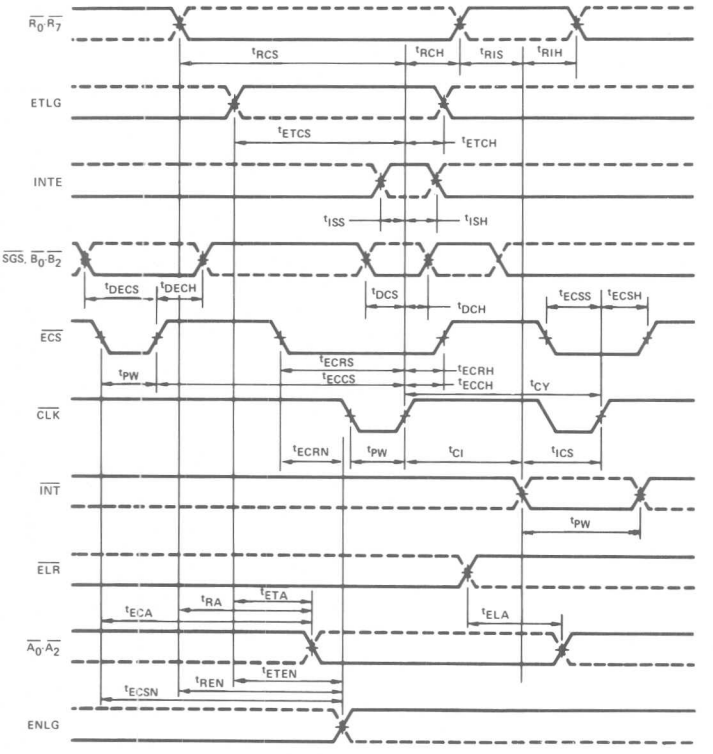
Capacitance[5]

Symbol	Parameter	Limits Min.	Typ.[1]	Max	Unit
C_{IN}	Input Capacitance		5	10	pF
C_{OUT}	Output Capacitance		7	12	pF

TEST CONDITIONS: $V_{BIAS} = 2.5V$, $V_{CC} = 5V$, $T_A = 25°C$, f = 1 MHz

NOTE 5. This parameter is periodically sampled and not 100% tested.

Waveforms



NOTES:

(1) Typical values are for $T_A = 25°C$, $V_{CC} = 5.0V$.

(2) Required for proper operation if ISE is enabled during next clock pulse.

(3) These times are not required for proper operation but for desired change in interrupt flip-flop.

(4) Required for new request or status to be properly loaded.

TEST CONDITIONS:

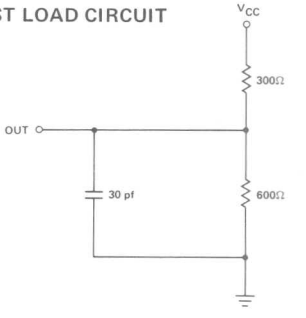
Input pulse amplitude: 2.5 volts.

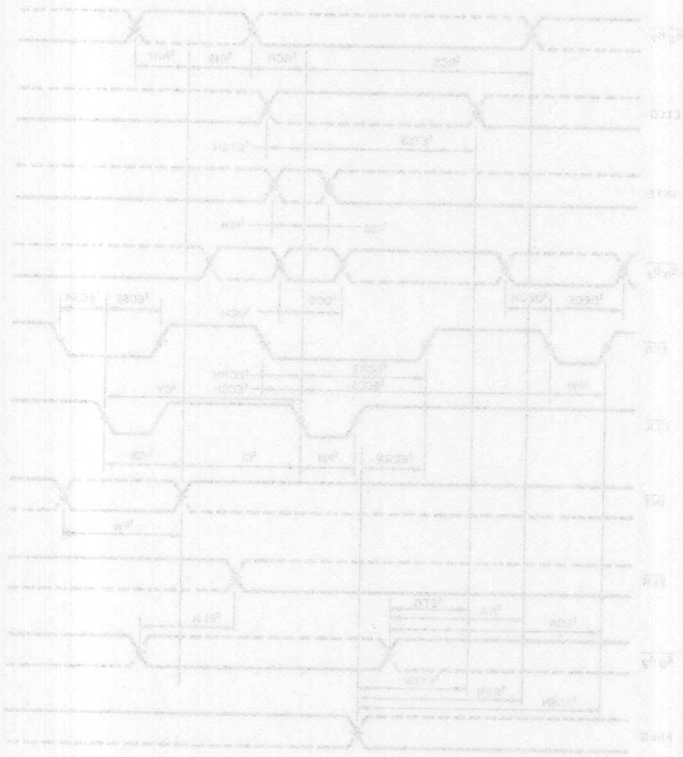
Input rise and fall times: 5 ns between 1 and 2 volts.

Output loading of 15 mA and 30 pf.

Speed measurements taken at the 1.5V levels.

TEST LOAD CIRCUIT





4 BIT PARALLEL BI-DIRECTIONAL BUS DRIVER

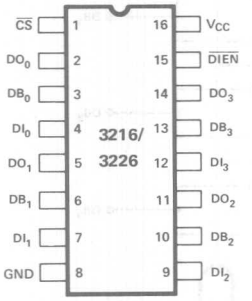
- Data Bus Buffer Driver
- Low Input Load Current: .25mA Maximum
- High Output Drive Capability for Driving System Data Bus
- 3.65V Output High Voltage
- Three-State Outputs
- Reduces System Package Count

The 3216/3226 is a 4-bit bi-directional bus driver/receiver.

All inputs are low power TTL compatible. For driving MOS, the DO outputs provide a high 3.65V V_{OH}, and for high capacitance terminated bus structures, the DB outputs provide a high 50mA I_{OL} capability.

A non-inverting (3216) and an inverting (3226) are available to meet a wide variety of applications for buffering in microcomputer systems.

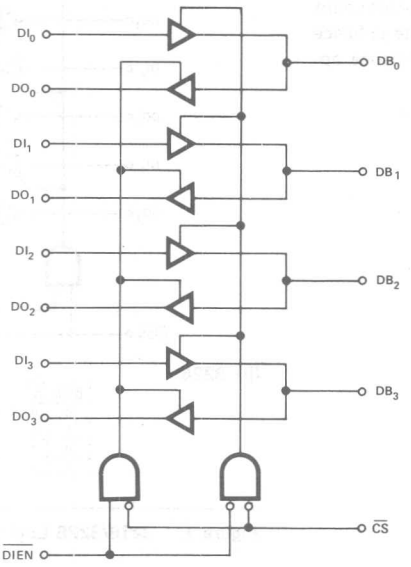
PIN CONFIGURATION



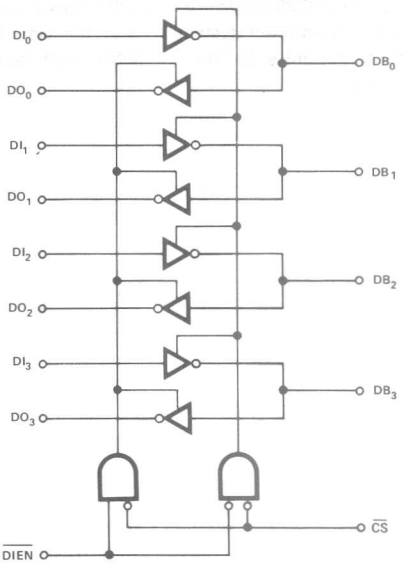
PIN NAMES

DB_0-DB_3	DATA BUS BI-DIRECTIONAL
DI_0-DI_3	DATA INPUT
DO_0-DO_3	DATA OUTPUT
$\overline{DIEN}$	DATA IN ENABLE DIRECTION CONTROL
$\overline{CS}$	CHIP SELECT

LOGIC DIAGRAM 3216



LOGIC DIAGRAM 3226



Functional Description

The 3216/3226 is a four bit bi-directional bus driver specifically designed to buffer microcomputer system components.

Bi-Directional Driver

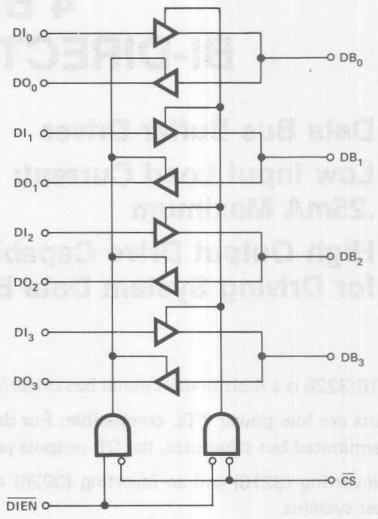
Each buffered line of the four bit driver consists of two separate buffers that are tri-state in nature to achieve direct bus interface and bi-directional capability. On one side of the driver the output of one buffer and the input of another are tied together (DB), this side is used to interface to the system side components such as memories, I/O, etc., because its interface is direct TTL compatible and it has high drive (50mA). On the other side of the driver the inputs and outputs are separated to provide maximum flexibility. Of course, they can be tied together so that the driver can be used to buffer a true bi-directional bus.

Control Gating $\overline{DIEN}$, $\overline{CS}$

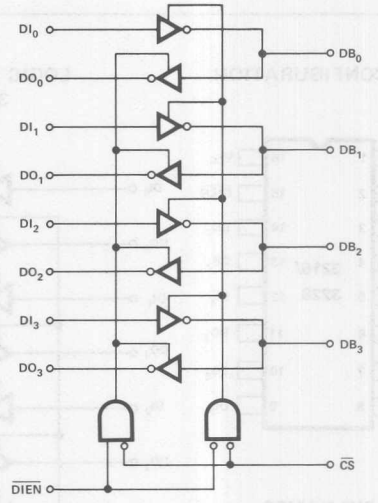
The $\overline{CS}$ input is actually a device select. When it is "high" the output drivers are all forced to their high-impedance state. When it is at "zero" the device is selected (enabled) and the direction of the data flow is determined by the $\overline{DIEN}$ input.

The $\overline{DIEN}$ input controls the direction of data flow (see Figure 1 for complete truth table). This direction control is accomplished by forcing one of the pair of buffers into its high impedance state and allowing the other to transmit its data. A simple two gate circuit is used for this function.

The 3216/3226 is a device that will reduce component count in microcomputer systems and at the same time enhance noise immunity to assure reliable, high performance operation.



(a) 3216



(b) 3226

$\overline{DIEN}$	$\overline{CS}$	
0	0	DI → DB
1	0	DB → DO
0	1	HIGH IMPEDANCE
1	1	

Figure 1. 3216/3226 Logic Diagrams

Absolute Maximum Ratings*

Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . -65°C to +150°C
All Output and Supply Voltages . . . . . -0.5V to +7V
All Input Voltages . . . . . -1.0V to +5.5V
Output Currents . . . . . 125 mA

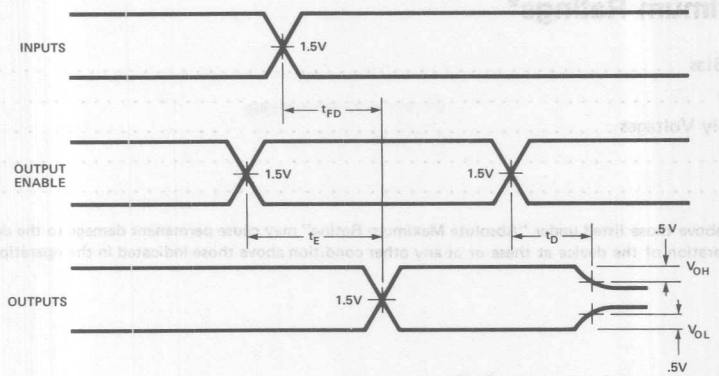
*COMMENT: Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics T_A = 0°C to +70°C, V_{CC}=+5V ±5%

Symbol	Parameter		Limits Min.	Typ.	Max.	Unit	Conditions
I_{F1}	Input Load Current $\overline{DIEN}$, $\overline{CS}$			-0.15	-.5	mA	V_F = 0.45
I_{F2}	Input Load Current All Other Inputs			-0.08	-.25	mA	V_F = 0.45
I_{R1}	Input Leakage Current $\overline{DIEN}$, $\overline{CS}$				20	μA	V_R = 5.25V
I_{R2}	Input Leakage Current DI Inputs				10	μA	V_R = 5.25V
V_C	Input Forward Voltage Clamp				-1	V	I_C = -5mA
V_{IL}	Input "Low" Voltage				.95	V	
V_{IH}	Input "High" Voltage		2.0			V	
$\|I_O\|$	Output Leakage Current (3-State)	DO DB			20 100	μA	V_O = 0.45V/5.25V
I_{CC}	Power Supply Current	3216		95	130	mA	
		3226		85	120	mA	
V_{OL1}	Output "Low" Voltage			0.3	.45	V	DO Outputs I_{OL}=15mA DB Outputs I_{OL}=25mA
V_{OL2}	Output "Low" Voltage	3216		0.5	.6	V	DB Outputs I_{OL}=55mA
		3226		0.5	.6	V	DB Outputs I_{OL}=50mA
V_{OH1}	Output "High" Voltage		3.65	4.0		V	DO Outputs I_{OH} = -1mA
V_{OH2}	Output "High" Voltage		2.4	3.0		V	DB Outputs I_{OH} = -10mA
I_{OS}	Output Short Circuit Current		-15 -30	-35 -75	-65 -120	mA mA	DO Outputs $V_O \cong 0V$, DB Outputs V_{CC}=5.0V

NOTE: Typical values are for T_A = 25°C, V_{CC} = 5.0V.

Waveforms



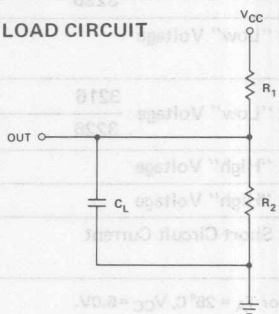
A.C. Characteristics $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$

Symbol	Parameter		Limits Min.	Typ.[1]	Max.	Unit	Conditions
T_{PD1}	Input to Output Delay DO Outputs			15	25	ns	C_L=30pF, R_1=300Ω R_2=600Ω
T_{PD2}	Input to Output Delay DB Outputs						
		3216		20	30	ns	C_L=300pF, R_1=90Ω
		3226		16	25	ns	R_2 = 180Ω
T_E	Output Enable Time						
		3216		45	65	ns	(Note 2)
		3226		35	54	ns	(Note 3)
T_D	Output Disable Time			20	35	ns	(Note 4)

TEST CONDITIONS:

Input pulse amplitude of 2.5V.
Input rise and fall times of 5 ns between 1 and 2 volts.
Output loading is 5 mA and 10 pF.
Speed measurements are made at 1.5 volt levels.

TEST LOAD CIRCUIT



Capacitance[5]

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Unit
C_{IN}	Input Capacitance		4	8	pF
C_{OUT1}	Output Capacitance		6	10	pF
C_{OUT2}	Output Capacitance		13	18	pF

TEST CONDITIONS: $V_{BIAS} = 2.5V$, $V_{CC} = 5.0V$, $T_A = 25°C$, f = 1 MHz.

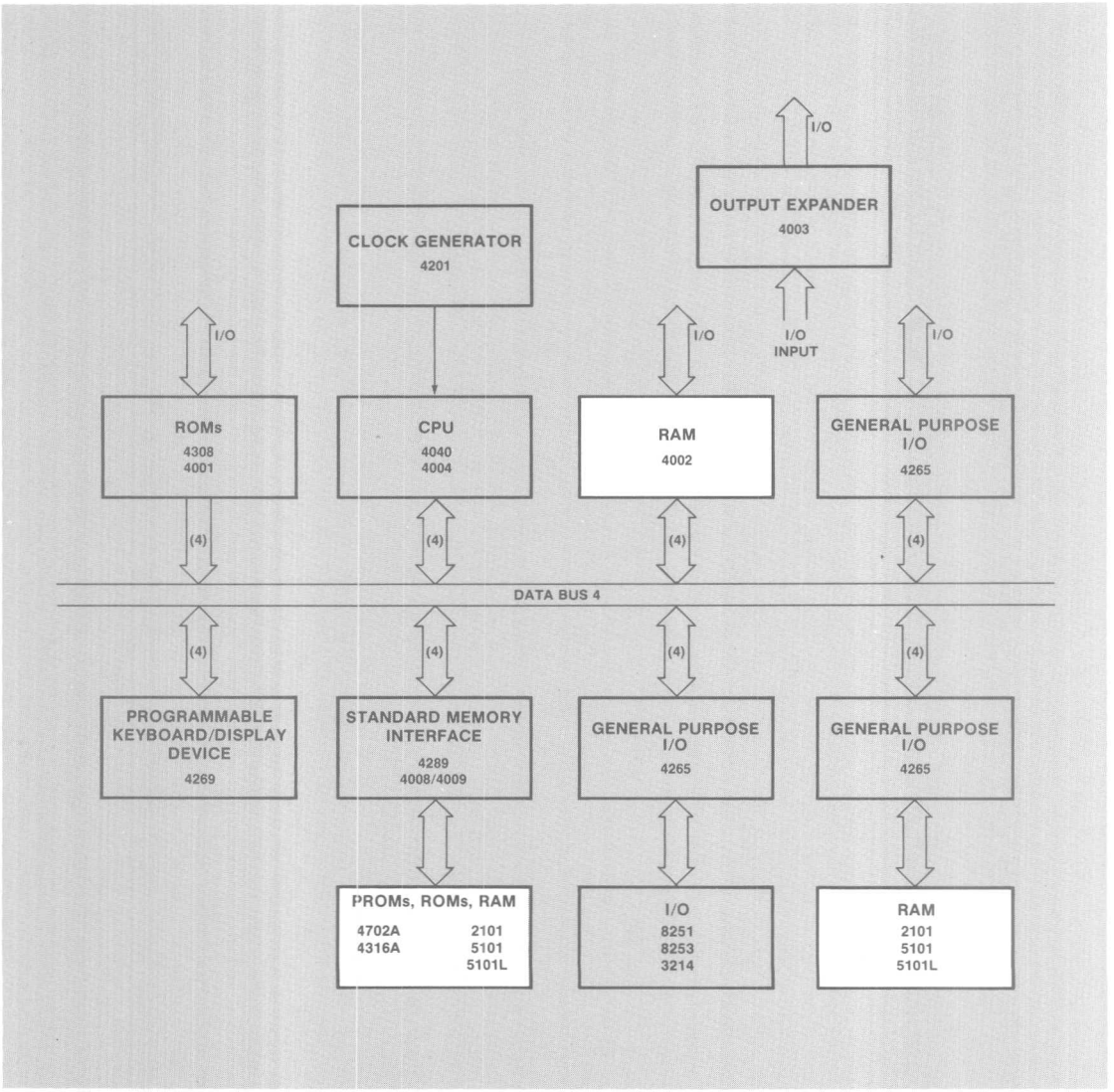
NOTES:
1. Typical values are for $T_A = 25°C$, $V_{CC} = 5.0V$.
2. DO Outputs, C_L = 30pF, R_1 = 300/10 KΩ, R_2 = 180/1KΩ; DB Outputs, C_L = 300pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
3. DO Outputs, C_L = 30pF, R_1 = 300/10 KΩ, R_2 = 600/1K; DB Outputs, C_L = 300pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
4. DO Outputs, C_L = 5pF, R_1 = 300/10 KΩ, R_2 = 600/1 KΩ; DB Outputs, C_L = 5pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
5. This parameter is periodically sampled and not 100% tested.

intel® Microcomputer Systems

RAMs

4002 5101

2101 5101L



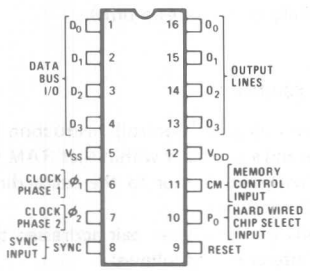
320 BIT RAM AND 4 BIT OUTPUT PORT

- Four Registers of 20 4 Bit Characters
- Direct Interface to MCS-40™ 4 Bit Bus
- Output Port Low-Power TTL Compatible
- 16 Pin Dual In-Line Package
- Standard Operating Temperature Range of 0° to 70°C
- Also Available With -40° to +85°C Operating Range

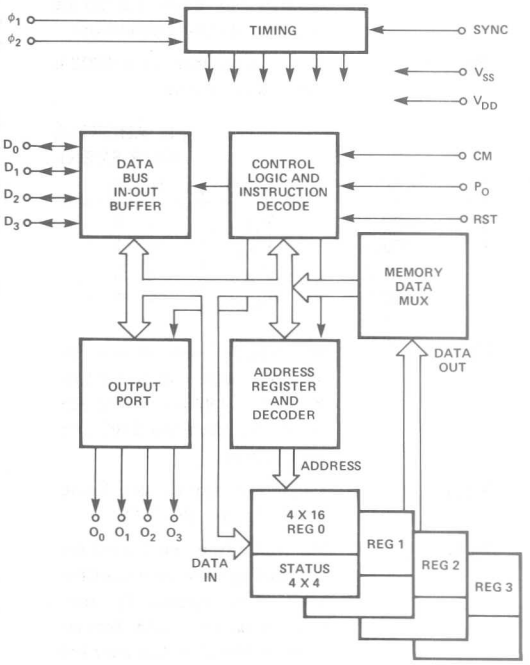
The 4002 performs two distinct functions. As a RAM it stores 320 bits arranged in 4 registers of twenty 4 bit characters each (16 main memory characters and 4 status characters). As a vehicle of communication with peripheral devices, it is provided with 4 output lines and associated control logic to perform output operations. The 4002 is a PMOS device and is compatible with all MCS-40™ components.

The 4002 is available in two options, the 4002-1 and 4002-2. Along with an external pin connected to either V_{DD} or V_{SS}, a two bit chip selection address is provided allowing a maximum of 1280 bits of 4002 RAM on a single MCS-40 CM-RAM line. Thus, the four CM-RAM lines give a maximum of 5120 bits of 4002 RAM in an MCS-40 system.

PIN CONFIGURATION



BLOCK DIAGRAM



Pin Description

Pin No.	Designation	Description of Function
1-4	D_0-D_3	Bidirectional data bus. All address, instruction and data communication between processor and the RAM MEMORY or the output port is transmitted on these 4 pins.
5	V_{SS}	Most positive supply voltage.
6-7	ϕ_1-ϕ_2	Non-overlapping clock signals which are used to generate the basic chip timing.
8	SYNC	Synchronization input signal driven by SYNC output of processor.
9	RESET	RESET input. A logic negative level (V_{DD}) applied to the chip will cause a clear of all output and control static flip-flops and will clear the RAM array. To completely clear the memory, RESET must be applied for at least 32 instruction cycles (256 clock periods) to allow the internal refresh counter to scan the memory. During RESET the data bus output buffers are inhibited (floating condition).
10	P_0	The chip number for a 4002 is assigned as follows:

Chip No.	4002 Option	P_0	SRC ADDRESS (RRR EVEN) D_3 D_2
0	4002-1	V_{SS}	0 0
1	4002-1	V_{DD}	0 1
2	4002-2	V_{SS}	1 0
3	4002-2	V_{DD}	1 1

11	CM	Command input driven by CM-RAM output of processor. Used for enabling the device during the decoding SRC and instructions.
12	V_{DD}	Main power supply pin. Value must be V_{SS} - 15V ± 5%.
13-16	O_3-O_0	Four bit output port used for transferring data from the CPU to the users system. The outputs are buffered and data remains stable after the port has been loaded. This port can be made low power TTL compatible by placing a 12K pull-down resistor to V_{DD} on each pin.

Functional Description

The twenty 4 bit characters for each 4002 register are arranged as follows:

1. 16 characters addressable by an SRC instruction. Four 16 character registers constitute the "main" memory.
2. 4 characters addressable by specific RAM instructions. Four 4 character registers constitute the "status character" memory.

The status character location (0 through 3) as well as the operation to be performed on it are selected by the OPA portion of the I/O and RAM instructions.

The RAM Registers Locations, Status Characters, and Output Port are select and accessed with a corresponding RAM Instruction.

There can be up to four RAMS per RAM Bank (CM-RAM). There can be four RAM banks per system without decoding or 8 with decoding.

Bank switching is accomplished by the CPU after receiving a "DCL" (designated command line) instruction. Prior to execution of the DCL instruction the desired CM-RAM code has been stored in the accumulator (for example through an LDM instruction). During DCL the CM-RAM code is transferred from the accumulator to the CM-RAM register. The RAM bank is then selected starting with the next instruction.

If no DCL is executed prior to SRC, the CM-RAM_0 will automatically be activated at the X_2 state of the instruction cycle provided that RESET was applied at least once to the system (most likely at the start-up time).

Instruction Execution

An SRC (Send Register Control) instruction is executed to select a RAM and a character within that RAM (for a RAM read or write instruction) prior to the succeeding RAM or I/O instruction's execution.

The eight bits of the register pair addressed by the SRC instruction are interpreted as follows:

a. The first four bits sent out at X_2 time select one out of four chips and one out of four registers. The two higher order bits (D_3, D_2) select the chip and the two lower order bits (D_1, D_0) select the register.

b. The second 4-bits (X_3 time) select one 4-bit character out of 16. The address is stored in the address register of the selected chip (second 4 bits are not used for status character reads or writes or for I/O output instructions).

The following RAM and I/O output instructions are executed by the 4002.

1. RDM Read RAM character

   The content of the previously selected RAM main memory character is transferred to the accumulator. The 4 bit data in memory is unaffected.

2. RDO-3 Read RAM status characters 0-3

   The 4 bits of status characters 0-3 for the previously selected RAM register are transferred to the accumulator.

3. WRM Write accumulator into RAM character

   The accumulator content is written into the previously selected RAM main memory character location.

4. WRO-3 Write accumulator into RAM status characters 0-3

   The content of the accumulator is written into the RAM status characters 0-3 of the previously selected RAM register.

5. WMP Write memory port

   The content of the accumulator is transferred to the RAM output port of the previously selected RAM chip. The data is available on the output pins until a new WMP is executed on the same RAM chip. The content of the ACC and the carry/link are unaffected. (The LSB bit of the accumulator appears on O_0, Pin 16 of the 4002.)

6. ADM Add from memory with carry

   The content of the previously selected RAM main memory character is added to the accumulator with carry. The RAM character is unaffected.

7. SBM Subtract from memory with borrow

   The content of the previously selected RAM character is subtracted from the accumulator with borrow. The RAM character is unaffected.

Timing Considerations

Presence of CM-RAM during X_2 tells 4002's that an SRC instruction was received. For a given combination of data at X_2 on D_2, D_3, only the chip with the proper option and P_0 state will be ready for the I/O or RAM operation that follows.

When an I/O or RAM instruction is received by the CPU, the CPU will activate one CM-RAM line during M_2, in time for the 4002's to receive the OPA (2nd part of the instruction), which will specify the I/O or RAM operation to be performed.

In the I/O mode of operation, the selected 4002 chip (by SRC), after receiving the OPA of an I/O instruction (CM-RAM activated at M_2), will decode the instruction.

If the instruction is WMP, the data present on the data bus during $X_2 \cdot \phi_2$ will set the output flip-flops associated with the I/O pins. That information will be available until next WMP for peripheral devices control.

In the RAM mode, the operation is as follows: When the CPU receives an SRC instruction, it will send out the content of the designated index register pair during X_2 and X_3 and will activate one CM-RAM line at X_2 for the previously selected RAM bank.

All RAM mode instructions will be executed during the X_2 and X_3. The instruction decoding is performed during the M_2 time when the OPA portion of the instruction is decoded. The CM-RAM of the selected Bank is enabled at that time.

Absolute Maximum Ratings*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	-55°C to + 125°C
Input Voltages and Supply Voltage with respect to Vss	+0.5V to -20V
Power Dissipation	1.0 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

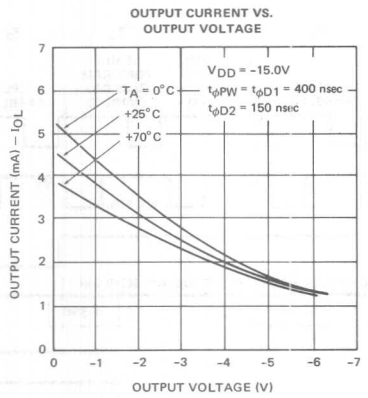
D.C. Characteristics

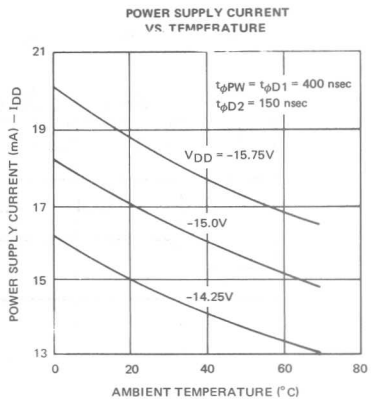
T_A = 0°C to 70°C; V_{SS}-V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec; $t_{\phi D2}$ = 150 nsec. Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless otherwise specified.

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
SUPPLY CURRENT						
I_{DD}	Average Supply Current		17	33	mA	T_A = 25°C
INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	V_{IL}=V_{DD}
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS – ALL OUTPUTS EXCEPT I/O PINS						
I_{LO}	Data Bus Output Leakage Current			10	μA	V_{OUT}=-12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitive Load
I_{OL}	Data Lines Sinking Current	8	15		mA	V_{OUT}=V_{SS}
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL}=0.5mA
R_{OH}	Output Resistance, Data Line "0" Level		150	250	Ω	V_{OUT}=V_{SS}-.5V
I/O OUTPUT CHARACTERISTICS						
V_{OH}	Output High Voltage	V_{SS}-.5V			V	I_{OUT}=0
R_{OH}	I/O Output "0" Resistance		1.2	2	kΩ	V_{OUT}=V_{SS}-.5V
I_{OL}	I/O Output "1" Sink Current	2.5	5		mA	V_{OUT}=V_{SS}-.5V
I_{OL}[1]	I/O Output "1" Sink Current	0.8	3		mA	V_{OUT}=V_{SS}-4.85V
V_{OL}	I/O Output Low Voltage	V_{SS}-12		V_{SS}-6.5	V	I_{OUT}=50μA
CAPACITANCE						
C_{ϕ}	Clock Capacitance		8	15	pF	V_{IN}=V_{SS}
C_{DB}	Data Bus Capacitance		7	10	pF	V_{IN}=V_{SS}
C_{IN}	Input Capacitance			10	pF	V_{IN}=V_{SS}
C_{OUT}	Output Capacitance			10	pF	V_{IN}=V_{SS}

Note: 1. For TTL compatibility, use 12kΩ external resistor to V_{DD}.

Typical D.C. Characteristics





A.C. Characteristics

T_A = 0°C to 70°C, V_{SS}–V_{DD} = 15V ±5%.

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Test Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t_{\phi R}$	Clock Rise Time			50	ns	
$t_{\phi F}$	Clock Fall Times			50	ns	
$t_{\phi PW}$	Clock Width	380		480	ns	
$t_{\phi D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t_{\phi D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT}=20pF
t_D	I/O Output Delay			1500	ns	C_{OUT}=100pF

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.

2. T_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.

3. All MCS-40 components which may transmit instruction or data to 4004/4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.

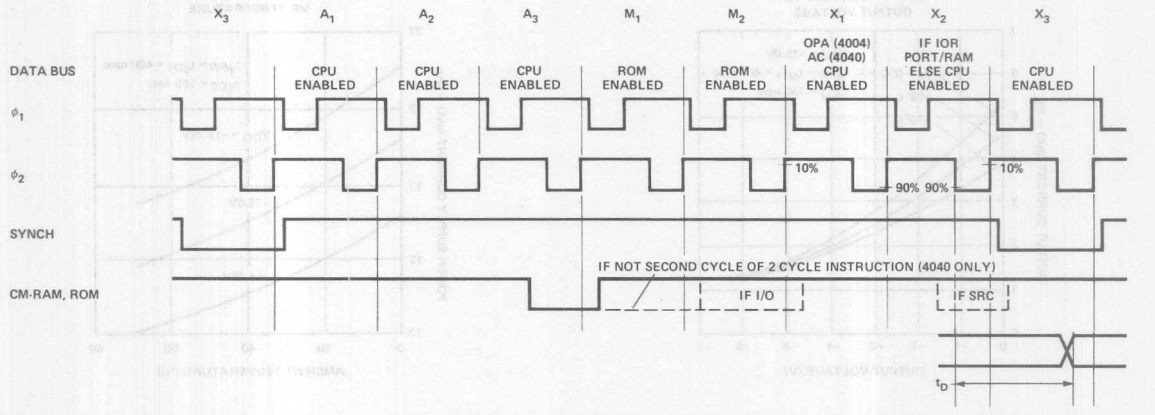


Figure 1. Timing Diagram.

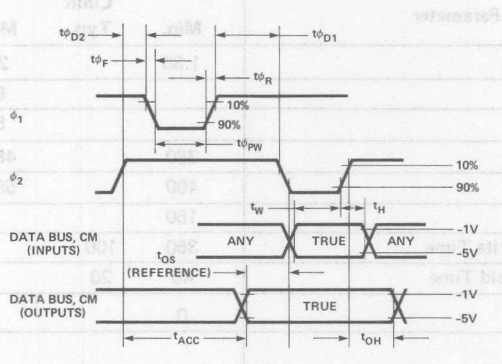


Figure 2. Timing Detail.

1024 BIT (256 x 4) STATIC MOS RAM WITH SEPARATE I/O

- 256 x 4 Organization to Meet Needs for Small System Memories
- Access Time: 1 µsec Max.
- Single +5V Supply Voltage
- Directly TTL Compatible: All Inputs and Outputs
- Statis MOS: No Clocks or Refreshing Required
- Simple Memory Expansion: Chip Enable Input
- Compatible with the 4289
- Inputs Protected: All Inputs Have Protection Against Static Charge
- Low Cost Packaging: 22 Pin Plastic Dual-In-Line Configuration
- Low Power: Typically 150mW
- Three-State Output: OR-Tie Capability
- Output Disable Provided for Ease of Use in Common Data Bus Systems

The Intel® 2101 is a 256 word by 4 bit static random access memory element using normally off N-channel MOS devices integrated on a monolithic array. It uses fully DC stable (static) circuitry and therefore requires no clocks or refreshing to operate. The data is read out nondestructively and has the same polarity as the input data.

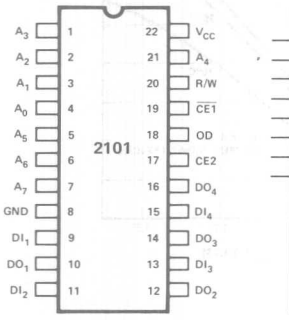
The 2101 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. Two chip-enables allow easy selection of an individual package when outputs are OR-tied. An output disable is provided so that data inputs and outputs can be tied for common I/O systems. Output disable is then used to eliminate any bi-directional logic.

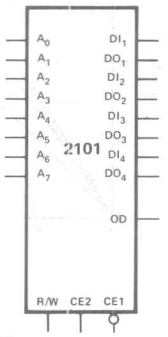
The Intel® 2101 is fabricated with N-channel silicon gate technology. This technology allows the design and production of high performance, easy-to-use MOS circuits and provides a higher functional density on a monolithic chip than either conventional MOS technology or P-channel silicon gate technology.

Intel's silicon gate technology also provides excellent protection against contamination. This permits the use of low cost plastic packaging.

PIN CONFIGURATION



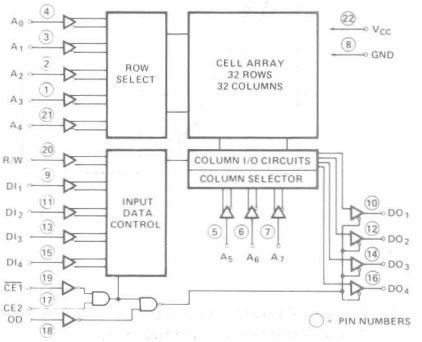
LOGIC SYMBOL



PIN NAMES

D_{IN}	DATA INPUT	OD	OUTPUT DISABLE
$A_0 - A_7$	ADDRESS INPUTS	D_{OUT}	DATA OUTPUT
R/W	READ/WRITE INPUT	V_{CC}	POWER (+5V)
$\overline{CE1}$, CE2	CHIP ENABLE		

BLOCK DIAGRAM



Absolute Maximum Ratings*

Ambient Temperature Under Bias . . . . . 0°C to 70°C

Storage Temperature . . . . . -65°C to +150°C

Voltage On Any Pin
With Respect to Ground . . . . . -0.5V to +7V

Power Dissipation . . . . . 1 Watt

*COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

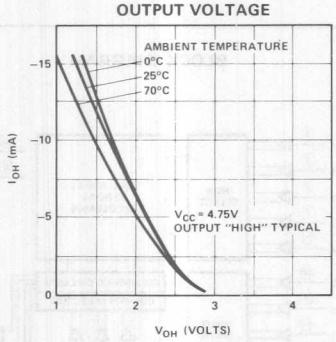
D.C. and Operating Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified.

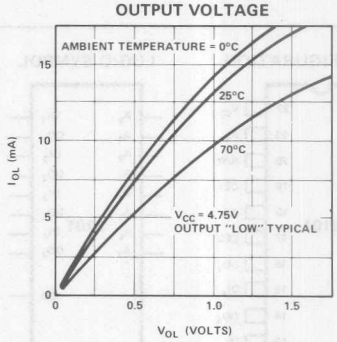
Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
I_{LI}	Input Current			10	μA	V_{IN} = 0 to 5.25V
I_{LOH}	I/O Leakage Current[2]			15	μA	$\overline{CE}_1$ = 2.2V, V_{OUT} = 4.0V
I_{LOL}	I/O Leakage Current[2]			-50	μA	$\overline{CE}_1$ = 2.2V, V_{OUT} = 0.45V
I_{CC1}	Power Supply Current		30	60	mA	V_{IN} = 5.25V, I_O = 0mA T_A = 25°C
I_{CC2}	Power Supply Current			70	mA	V_{IN} = 5.25V, I_O = 0mA T_A = 0°C
V_{IL}	Input "Low" Voltage	-0.5		+0.65	V	
V_{IH}	Input "High" Voltage	2.2		V_{CC}	V	
V_{OL}	Output "Low" Voltage			+0.45	V	I_{OL} = 2.0mA
V_{OH}	Output "High" Voltage	2.2			V	I_{OH} = -150 μA

Typical D.C. Characteristics

OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltage.
2. Input and Output tied together.

A.C. Characteristics

READ CYCLE $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$, unless otherwise specified.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{RC}	Read Cycle	1,000			ns	t_r, t_f = 20ns V_{IN} = +0.65V to +2.2V Timing Reference = 1.5V Load = 1 TTL Gate and C_L = 100pF.
t_A	Access Time			1,000	ns	
t_{CO}	Chip Enable To Output			800	ns	
t_{OD}	Output Disable To Output			700	ns	
t_{DF}[3]	Data Output to High Z State	0		200	ns	
t_{OH}	Previous Read Data Valid after change of Address	40			ns	

WRITE CYCLE

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{WC}	Write Cycle	1,000			ns	t_r, t_f = 20ns V_{IN} = +0.65V to +2.2V Timing Reference = 1.5V Load = 1 TTL Gate and C_L = 100pF.
t_{AW}	Write Delay	150			ns	
t_{CW}	Chip Enable To Write	900			ns	
t_{DW}	Data Setup	700			ns	
t_{DH}	Data Hold	100			ns	
t_{WP}	Write Pulse	750			ns	
t_{WR}	Write Recovery	50			ns	
t_{DS}	Output Disable Setup	200			ns	

A. C. CONDITIONS OF TEST

Input Pulse Levels: +0.65 Volt to 2.2 Volt

Input Pulse Rise and Fall Times: 20nsec

Timing Measurement Reference Level: 1.5 Volt

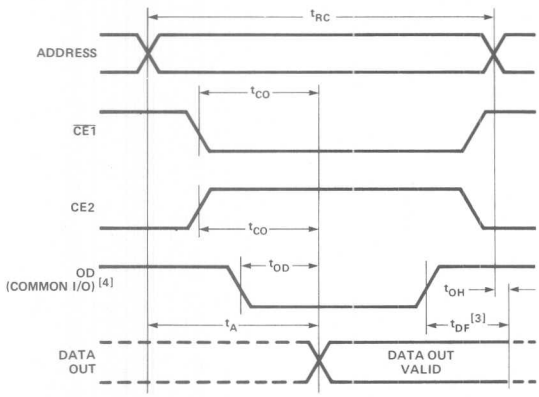
Output Load: 1 TTL Gate and C_L = 100pF

Capacitance[2] $T_A = 25°C$, f = 1MHz

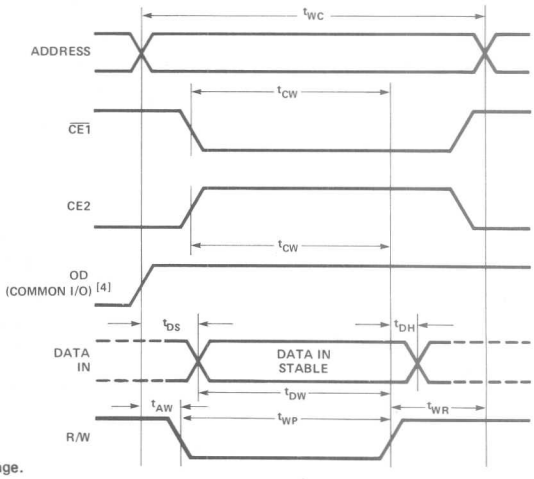
Symbol	Test	Limits (pF)	
		Typ.[1]	Max.
C_{IN}	Input Capacitance (All Input Pins) V_{IN} = 0V	4	8
C_{OUT}	Output Capacitance V_{OUT} = 0V	8	12

Waveforms

READ CYCLE



WRITE CYCLE



NOTES:
1. Typical values are for $T_A = 25°C$ and nominal supply voltage.
2. This parameter is periodically sampled and is not 100% tested.
3. t_{DF} is with respect to the trailing edge of $\overline{CE_1}$, CE_2, or OD, whichever occurs first.
4. OD should be tied low for separate I/O operation.

A.C. Characteristics

READ CYCLE T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{RC}	Read Cycle	1,000			ns	
t_A	Access Time			1,000	ns	t_r, t_f = 20ns
t_{OD}	Chip Enable To Output			800	ns	V_{IH} = +0.65V to +2.2V
t_{OD}	Output Disable To Output			200	ns	Timing Reference = 1.5V
t_{DF}[2]	Data Output to High Z State	0		200	ns	Load = 1 TTL Gate
t_{OH}	Previous Read Data Valid after change of Address	40			ns	and C_L = 100pF.

WRITE CYCLE

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{WC}	Write Cycle	1,000			ns	
t_{AW}	Write Delay	150			ns	t_r, t_f = 20ns
t_{CW}	Chip Enable To Write	900			ns	V_{IH} = +0.65V to +2.2V
t_{DW}	Data Setup	700			ns	Timing Reference = 1.5V
t_{DH}	Data Hold	100			ns	Load = 1 TTL Gate
t_{WP}	Write Pulse	750			ns	and C_L = 100pF.
t_{WR}	Write Recovery	50			ns	
t_{OS}	Output Disable Setup	200			ns	

A. C. CONDITIONS OF TEST

Input Pulse Levels: +0.65 Volt to 2.2 Volt

Input Pulse Rise and Fall Times: 20nsec

Timing Measurement Reference Level: 1.5 Volt

Output Load: 1 TTL Gate and C_L = 100pF

Capacitance[3] T_A = 25°C, f = 1MHz

Symbol	Test	Limits (pF) Typ.[1]	Max.
C_{IN}	Input Capacitance (All Input Pins at 0V)	4	8
C_{OUT}	Output Capacitance V_{OUT} = 0V	8	12

Waveforms

READ CYCLE

WRITE CYCLE

NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltage.
2. This parameter is periodically sampled and is not 100% tested.
3. t_{DF} is with respect to the trailing edge of $\overline{CE}_1$, CE_2, or $\overline{OD}$, whichever occurs first.
4. OD should be tied to Chip Enable for separate I/O operation.

5101, 5101L

1024 BIT (256 x 4) STATIC CMOS RAM

P/N	Typ. Current @ 2V (μA)	Typ. Standby Current (μA)	Max Access (ns)
5101L	0.14	0.2	650
5101L-1	0.9	1.5	450
5101L-3	0.7	1.0	650
5101-1	—	1.5	450
5101	—	0.2	650
5101-3	—	1.0	650
5101-8	—	10.0	800

- **Single +5V Power Supply**
- **Ideal for Battery Operation (5101L)**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Three-State Output**

The Intel® 5101 and 5101L are ultra-low power 1024 bit (256 words x 4-bits) static RAMs fabricated with an advanced ion-implanted silicon gate CMOS technology. The devices have two chip enable inputs. Minimum standby current is drawn by these devices when CE_2 is at a low level. When deselected the 5101 draws from the single 5 volt supply only 15 microamps. These devices are ideally suited for low power applications where battery operation or battery backup for non-volatility are required.

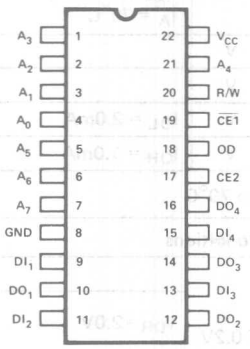
The 5101 uses fully DC stable (static) circuitry; it is not necessary to pulse chip select for each address transition. The data is read out non-destructively and has the same polarity as the input data. All inputs and outputs are directly TTL compatible. The 5101 has separate data input and data output terminals. An output disable function is provided so that the data inputs and outputs may be wire OR-ed for use in common data I/O systems.

The 5101L is identical to the 5101 with the additional feature of guaranteed data retention at a power supply voltage as low as 2.0 volts.

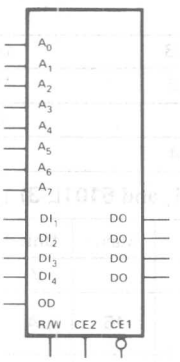
A pin compatible N-channel static RAM, the Intel® 2101, is also available for low cost applications where a 256 x 4 organization is needed.

The Intel ion-implanted, silicon gate, complementary MOS (CMOS) allows the design and production of ultra-low power, high performance memories.

PIN CONFIGURATION



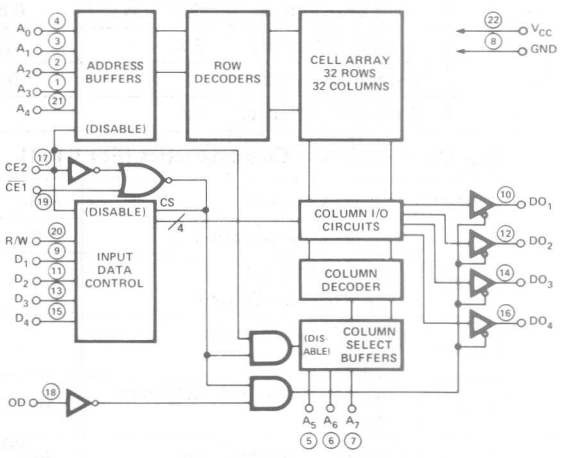
LOGIC SYMBOL



TRUTH TABLE

$\overline{CE_1}$	CE_2	OD	R/W	D_{IN}	Output	Mode
H	X	X	X	X	High Z	Not Selected
X	L	X	X	X	High Z	Not Selected
X	X	H	H	X	High Z	Output Disabled
L	H	H	L	X	High Z	Write
L	H	L	L	X	D_{IN}	Write
L	H	L	H	X	D_{OUT}	Read

BLOCK DIAGRAM



Absolute Maximum Ratings *

Ambient Temperature Under Bias . . . . –10°C to 80°C
Storage Temperature . . . . . -65°C to +150°C
Voltage On Any Pin
With Respect to Ground . . . . -0.3V to V_{CC} +0.3V
Maximum Power Supply Voltage . . . . . +7.0V
Power Dissipation . . . . . 1 Watt

*COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D. C. and Operating Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified.

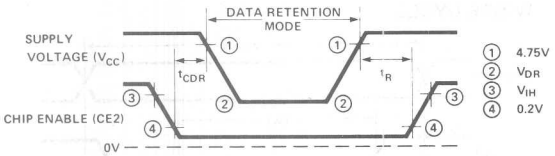
Symbol	Parameter	5101 (Except 5101-8) and 5101L Family Limits Min.	Typ.[1]	Max.	5101-8 Limits Min.	Typ.[1]	Max.	Units	Test Conditions
I_{LI}[2]	Input Current		5			5		nA	
$\|I_{LO}\|$[2]	Output Leakage Current			1			2	μA	$\overline{CE1}$ = 2.2V, V_{OUT} = 0 to V_{CC}
I_{CC1}	Operating Current		9	22		11	25	mA	V_{IN} = V_{CC}, Except $\overline{CE1} \leqslant$ 0.65V, Outputs Open
I_{CC2}	Operating Current		13	27		15	30	mA	V_{IN} = 2.2V, Except $\overline{CE1} \leqslant$ 0.65V, Outputs Open
I_{CCL1}[2]	5101 and 5101-1 Standby Current			15			—	μA	CE2 ⩽ 0.2V, V_{CC} = 5V ±5%
I_{CCL2}[2]	5101-3 Standby Current		1	200			—	μA	CE2 ⩽ 0.2V, V_{CC} = 5V ±5%
I_{CCL3}[2]	5101-8 Standby Current			—		10	50	μA	CE2 ⩽ 0.2V, V_{CC} = 5V ±5%, T_A = 25°C
I_{CCL4}[2]	5101-8 Standby Current			—			500	μA	CE2 ⩽ 0.2V, V_{CC} = 5V ±5%, T_A = 70°C
V_{IL}	Input Low Voltage	-0.3		0.65	-0.3		0.65	V	
V_{IH}	Input High Voltage	2.2		V_{CC}	2.2		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4			0.4	V	I_{OL} = 2.0mA
V_{OH}	Output High Voltage	2.4			2.4			V	I_{OH} = 1.0mA

Low V_{CC} Data Retention Characteristics (For 5101L, 5101L-1, and 5101L-3) T_A = 0°C to 70°C

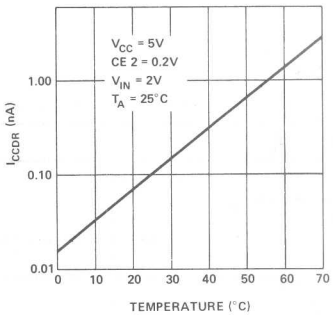
Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions	
V_{DR}	V_{CC} for Data Retention	2.0			V		
I_{CCDR1}	5101L or 5101L-1 Data Retention Current		0.14	15	μA	CE2 ⩽ 0.2V	V_{DR} = 2.0V
I_{CCDR2}	5101L-3 Data Retention Current		0.7	200	μA		V_{DR} = 2.0V
t_{CDR}	Chip Deselect to Data Retention Time	0			ns		
t_R	Operation Recovery Time	t_{RC}[3]			ns		

NOTES: 1. Typical values are T_A = 25°C and nominal supply voltage. 2. Current through all inputs and outputs included in I_{CCL} measurement. 3. t_{RC} = Read Cycle Time.

Low V_{CC} Data Retention Waveform



Typical I_{CCDR} Vs. Temperature



A.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified.

READ CYCLE

Symbol	Parameter	5101-1, 5101L-1 Limits (ns) Min.	Max.	5101, 5101-3, 5101L and 5101L-3 Limits (ns) Min.	Max.	5101-8 Limits (ns) Min.	Max.
t_{RC}	Read Cycle	450		650		800	
t_A	Access Time		450		650		800
t_{CO1}	Chip Enable ($\overline{CE\ 1}$) to Output		400		600		800
t_{CO2}	Chip Enable (CE 2) to Output		500		700		850
t_{OD}	Output Disable to Output		250		350		450
t_{DF}	Data Output to High Z State	0	130	0	150	0	200
t_{OH1}	Previous Read Data Valid with Respect to Address Change	0		0		0	
t_{OH2}	Previous Read Data Valid with Respect to Chip Enable	0		0		0	

WRITE CYCLE

t_{WC}	Write Cycle	450	650	800
t_{AW}	Write Delay	130	150	200
t_{CW1}	Chip Enable ($\overline{CE\ 1}$) to Write	350	550	650
t_{CW2}	Chip Enable (CE 2) to Write	350	550	650
t_{DW}	Data Setup	250	400	450
t_{DH}	Data Hold	50	100	100
t_{WP}	Write Pulse	250	400	450
t_{WR}	Write Recovery	50	50	100
t_{DS}	Output Disable Setup	130	150	200

A. C. CONDITIONS OF TEST

Input Pulse Levels: +0.65 Volt to 2.2 Volt

Input Pulse Rise and Fall Times: 20nsec

Timing Measurement Reference Level: 1.5 Volt

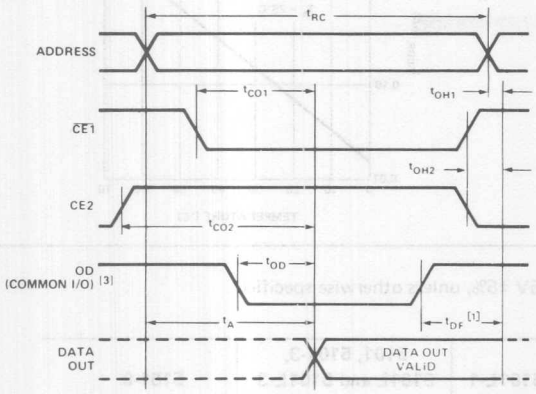
Output Load: 1 TTL Gate and C_L 100pF

Capacitance[2] T_A = 25°C, f = 1MHz

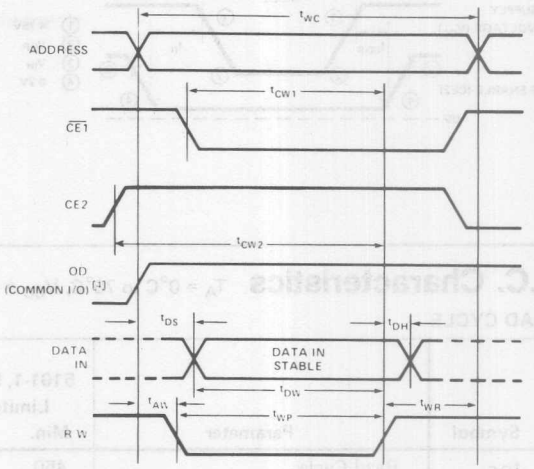
Symbol	Test	Limits (pF) Typ.	Max.
C_{IN}	Input Capacitance (All Input Pins) V_{IN} = 0V	4	8
C_{OUT}	Output Capacitance V_{OUT} = 0V	8	12

Waveforms

READ CYCLE



WRITE CYCLE

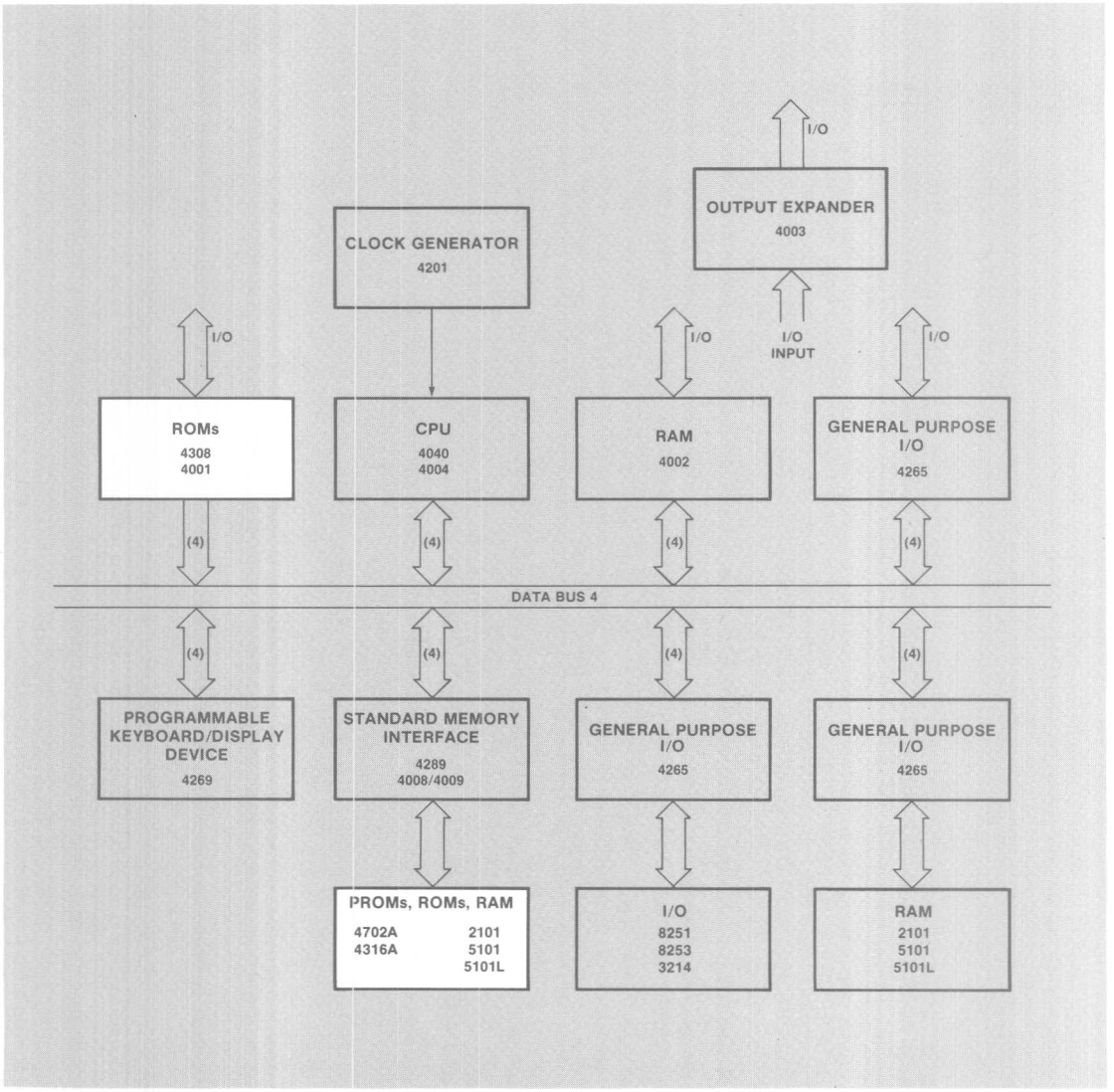


NOTES:
1. Typical values are for T_A = 25° C and nominal supply voltage.
2. This parameter is periodically sampled and is not 100% tested.
3. OD may be tied low for separate I/O operation.
4. During the write cycle, OD is "high" for common I/O and "don't care" for separate I/O operation.

intel Microcomputer Systems

ROMs
4001
4308
4316A

PROMs
4702A



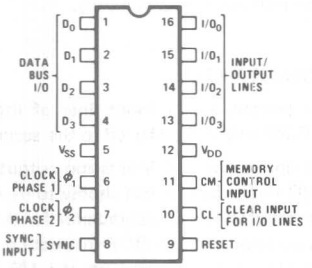
4001

256 x 8 MASK PROGRAMMABLE ROM AND 4 BIT I/O PORT

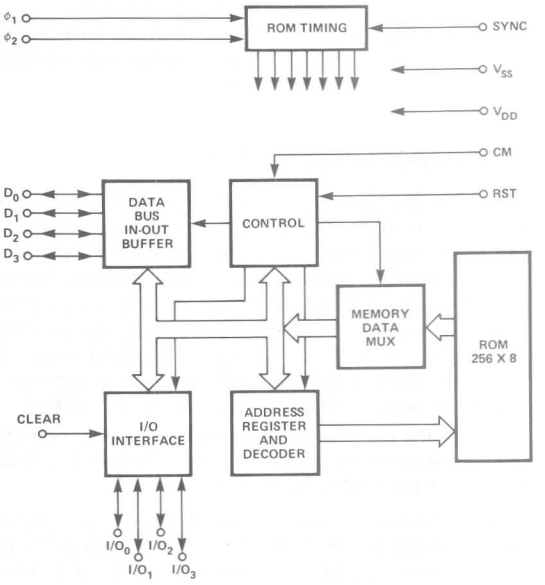
- **Direct Interface to MCS-40™ 4 Bit Data Bus**
- **I/O Port Low-Power TTL Compatible**
- **16 Pin Dual In-Line Package**
- **Standard Operating Temperature Range of 0° to 70° C**
- **Also Available With -40° to +85° C Operating Range**

The 4001 performs two basic and distinct functions. As a ROM it stores 256 x 8 words of program or data tables; as a vehicle of communication with peripheral devices it is provided with 4 I/O pins and associated control logic to perform input and output operations. The 4001 is a PMOS device, compatible with all other MCS-40™ devices.

PIN CONFIGURATION



BLOCK DIAGRAM



Pin Description

Pin No.	Designation/ Type of Logic	Description of Function
1-4	D_0 -D_3/Neg.	Bidirectional data bus. All address and data communication between the processor and ROM is handled by these lines.
5	V_{SS}	Most positive supply voltage.
6-7	$\phi1$, $\phi2$/Neg.	Non-overlapped clock signals which determine device timing.
8	SYNC/Neg.	System synchronization signal generated by processor.
9	RESET/Neg.	Reset input. A negative level (V_{DD}) on this pin will clear internal flip-flops and buffers. The input buffers are not cleared by this signal.
10	CL/Neg.	Clear input for I/O lines. A negative level on this pin will clear the I/O buffers. This pin may be driven by a TTL output and a 1K pull-up to V_{SS}.
11	CM-ROM/Neg.	Chip enable generated by the processor.
12	V_{DD}	Main supply voltage value. Must be V_{SS} – 15.0V ±5%.
13-16	I/O_0-I/O_3/Neg.	A single I/O port consisting of 4 bidirectional and selectable lines.

Functional Description

Address and data are transferred in and out by time multiplexing on 4 data bus lines. Timing is internally generated using two clock signals, ϕ_1 and ϕ_2, and a SYNC signal supplied by the CPU. Addresses are received from the CPU on three time periods following SYNC, and select 1 out of 256 words and 1 out of 16 ROM's. For that purpose, each ROM is identified as #0, 1, 2, through 15, by metal option. A Command ROM Line (CM-ROM) is also provided and it is used to select a ROM bank (group of 16 ROM's).

During the two time periods of the instruction cycle (M_1 & M_2) following the addressing time, information is transferred from the ROM to the data bus lines.

A second mode of operation of the ROM is as an Input/Output control device. In that mode a ROM chip will route information to and from data bus lines in and out of 4 I/O external lines. Each chip has the capability to identify itself for an I/O port operation, recognize an I/O port instruction and decide whether it is an Input or an Output operation and execute the instruction. An external signal (CL) will asynchronously clear the output register during normal operation.

All internal flip flops (including the output register) will be reset when the RESET line goes low (V_{DD}).

I/O Options

Each I/O pin on each ROM can be uniquely chosen to be either an input or output line by metal option. Also each input or output can either be inverted or direct. When the pin is chosen as an input it may have an on-chip resistor connected to either V_{DD} or V_{SS}.

Instruction Execution

The 4001 responds to the following instructions.

1. **SRC Instruction** (Send address to ROM and RAM)

   When the CPU executes an SRC instruction it will send out 8 bits of data during X_2 and X_3 and will activate the CM-ROM and one CM-RAM line at X_2. Data at X_2, (representing the contents of the first register of the register pair addressed by the SRC instruction) with simultaneous presence of CM-ROM, is interpreted by the 4001 as the chip number of the unit that should later perform an I/O operation. Data at X_3 is ignored.

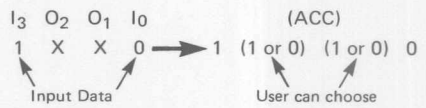
2. **WRR – Write ROM Port**

   The content of the accumulator is transferred to the ROM output port of the previously selected ROM chip. The data is available on the output pins until a new WRR is executed on the same chip. The ACC content and carry/link are unaffected. (The LSB bit of the accumulator appears on I/O_0.) No operation is performed on I/O lines coded as inputs.

3. **RDR – Read ROM Port**

   The data present at the input lines of the previously selected ROM chip is transferred to the accumulator.

   If the I/O option has both inputs and outputs within the same 4 I/O lines, the user can choose to have either "0" or "1" transferred to the accumulator for those I/O pins coded as outputs, when an RDR instruction is executed. For example, given a port with the I/O lines coded with 2 inputs and 2 outputs, when RDR is executed, the transfer is as shown below:



Timing Consideration

In the ROM mode of operation the 4001 will receive an 8 bit address during A_1 and A_2 times of the instruction cycle and a chip number, together with CM-ROM, during A_3 time. When CM-ROM is present, only the chip whose metal option code matches the chip number code sent during A_3 is allowed to send data out during the following two cycles: M_1 and M_2. The activity of the 4001 in the ROM mode ends at M_2.

The 4001 can have a chip number via the metal option from 0 – 15.

In the I/O mode of operation, the selected 4001 (by SRC), after receiving RDR will transfer the information present at its I/O pins to the data bus at X_2. If the instruction received was WRR, the data present on the data bus at $X_2 \cdot \phi_2$ will be latched on the output flip-flops associated with the I/O lines.

Ordering Information

When ordering a 4001, the following information must be specified:

1. Chip number
2. All the metal options for *each* I/O pin.
3. ROM pattern to be stored in each of the 256 locations.

A blank customer truth table is available upon request from Intel. A copy of this table is shown and blank copy can be found following the detailed 4001 characteristics.

EXAMPLES – DESIRED OPTION/CONNECTIONS REQUIRED

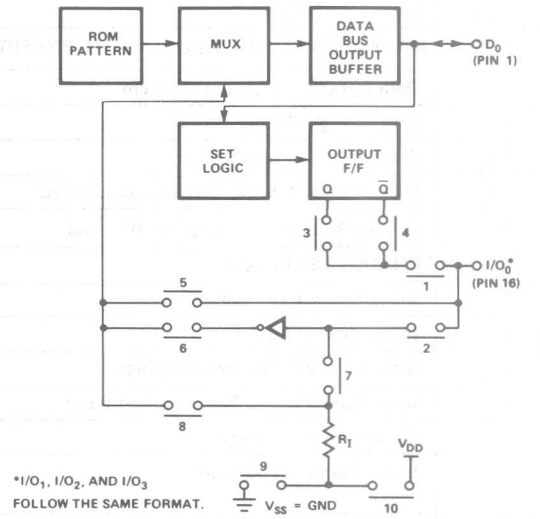
1. Non-inverting output (negative logic output) – 1 and 3 are connected.
2. Inverting output (positive logic output) – 1 and 4 are connected.
3. Non-inverting input (no input resistor – negative logic input) – only 5 is connected.
4. Inverting input (input resistor to V_{SS} – positive logic input) – 2, 6, 7, and 9 are connected.
5. Non-inverting input (input resistor to V_{DD} – negative logic input) – 2, 7, 8, and 10 are connected.
6. If inputs and outputs are mixed on the same port, the pins used as the outputs must have the internal resistor connected to either V_{DD} or V_{SS} (8 and 9 or 8 and 10 must be connected). This is necessary for testing purposes. For example, if there are two inverting inputs (with no input resistor) and 2 non-inverting outputs the connection would be made as follows:

   Inputs – 2 and 6 are connected
   Outputs – 1, 3, 8 and 9 are connected or
   1, 3, 8 and 10 are connected

   If the pins on a port are all inputs or all outputs, the internal resistors do not have to be connected.

It should be noted that all internal logic and processing is performed in negative logic, i.e., "1" equals V_{DD} and "0" equals V_{SS}. For positive logic conversion, the inverted options should be selected.

TTL compatibility is obtained by $V_{DD} = -10V \pm 5\%$ and $V_{SS} = 5V \pm 5\%$. An external 12K resistor should be used on all outputs to insure the logic "0" state (V_{OL}).



4001 Available Metal Option for Each I/O Pin.

Absolute Maximum Ratings*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	-55°C to + 125°C
Input Voltages and Supply Voltage with respect to V_{SS}	+0.5V to -20V
Power Dissipation	1.0 Watt

COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW} = t_{\phi D1}$ = 400 nsec; $t_{\phi D2}$ = 150 nsec; Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless Otherwise Specified.

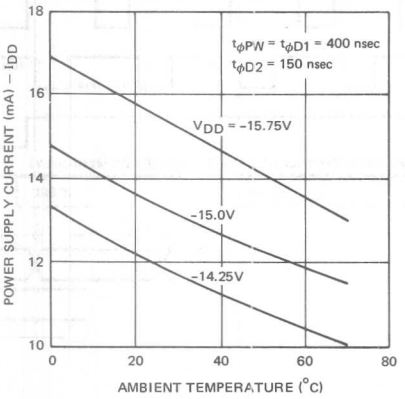
SUPPLY CURRENT

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
I_{DD}	Average Supply Current		15	30	mA	T_A = 25°C
INPUT CHARACTERISTICS – ALL INPUTS EXCEPT I/O PINS						
I_{LI}	Input Leakage Current			10	μA	$V_{IL} = V_{DD}$
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS – ALL OUTPUTS EXCEPT I/O PINS						
I_{LO}	Data Bus Output Leakage Current			10	μA	V_{OUT} = -12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitive Load
I_{OL}	Data Lines Sinking Current	8	15		mA	$V_{OUT} = V_{SS}$
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL} = 0.5mA
R_{OH}	Output Resistance, Data Line "0" Level		150	250	Ω	$V_{OUT} = V_{SS}$ -.5V
I/O INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	
V_{IH}	Input High Voltage	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage, Inverting Input	V_{DD}		V_{SS}-4.2	V	
V_{IL}	Input Low Voltage, Non-inverting Input	V_{DD}		V_{SS}-6.5	V	
V_{IL}	CL Input Low Voltage	V_{DD}		V_{SS}-4.2	V	
R_I	Input Resistance, if Used	10	18	35	kΩ	R_1 tied to V_{SS}; $V_{IN} = V_{SS}$ -3V
R_1[1]	Input Resistance, if Used	15	25	40	kΩ	R_1 tied to V_{DD}; $V_{IN} = V_{SS}$ -3V
I/O OUTPUT CHARACTERISTICS						
V_{OH}	Output High Voltage	V_{SS}-.5V			V	I_{OUT} = 0
R_{OH}	I/O Output "0" Resistance		1.2	2	kΩ	$V_{OUT} = V_{SS}$ -.5V
I_{OL}	I/O Output "1" Sink Current	2.5	5		mA	$V_{OUT} = V_{SS}$ -.5V
I_{OL}[2]	I/O Output "1" Sink Current	0.8	3		mA	$V_{OUT} = V_{SS}$ -4.85V
V_{OL}	I/O Output Low Voltage	V_{SS}-12		V_{SS}-6.5	V	I_{OUT} = 50μA
CAPACITANCE						
C_ϕ	Clock Capacitance		8	15	pF	$V_{IN} = V_{SS}$
C_{DB}	Data Bus Capacitance		9.5	15	pF	$V_{IN} = V_{SS}$
C_{IN}	Input Capacitance			10	pF	$V_{IN} = V_{SS}$
C_{OUT}	Output Capacitance			10	pF	$V_{IN} = V_{SS}$

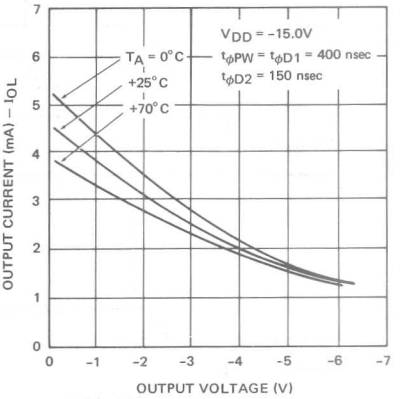
Notes: 1. R_I is large signal equivalent resistance to (V_{SS}-12) V.
2. For TTL compatibility, use 12kΩ external resistor to V_{DD}.

Typical D.C. Characteristics

POWER SUPPLY CURRENT VS. TEMPERATURE



OUTPUT CURRENT VS. OUTPUT VOLTAGE



A.C. Characteristics $T_A = 0°C$ to $70°C$, $V_{SS} - V_{DD} = 15V \pm 5\%$

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t\phi_R$	Clock Rise Time			50	ns	
$t\phi_F$	Clock Fall Times			50	ns	
$t\phi_{PW}$	Clock Width	380		480	ns	
$t\phi_{D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t\phi_{D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT} = 20pF
t_{IS}	I/O Input Set-Time	50			ns	
t_{IH}	I/O Input Hold-Time	100			ns	
t_D	I/O Output Delay			1500	ns	C_{OUT} = 100pF
t_C[4]	I/O Output Lines Delay on Clear			1500	ns	C_{OUT} = 100pF

Notes: 1. t_H measured with $t_{\phi R}$ = 10nsec.

2. T_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.

3. All MCS-40 components which may transmit instruction or data to 4004/4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.

4. CL on the 4001 is used to asynchronously clear the output flip-flops associated with the I/O lines.

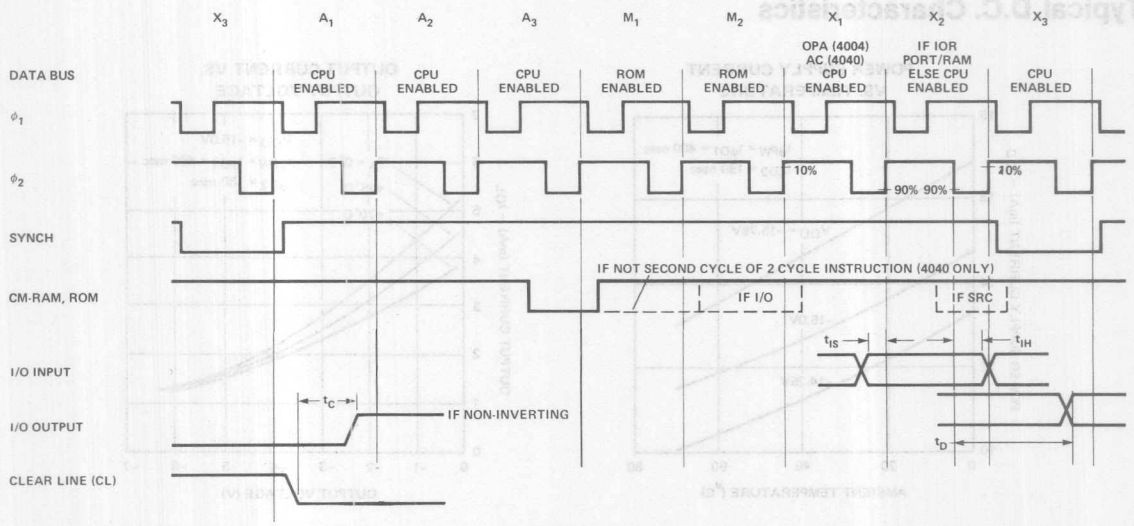


Figure 1. Timing Diagram

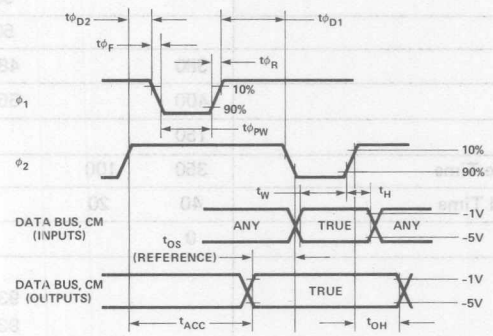


Figure 2. Timing Detail

Programming Instructions

To insure optimum handling of ROM programs and avoid delays, programs should be specified in the following format.

Paper Tape Format*

A 1" wide paper tape using 8 bit ASCII code, such as a model 33ASR teletype produces:

A. Preamble

1. Preceding the first word field and following the last word field, there should be a leader/trailer length of at least 25 characters. This should consist of rubout punches.

2. Included in the tape before the leader, and preceded by another leader, should be the customer's complete telex or twx number and if more than one pattern is being transmitted, the ROM pattern number.

3. The first ROM pattern preamble field is the device type number or ROM number. The field should be framed by an "I" and "–"

   I4001–

   This should be followed by the chip select information encoded in decimal (two digits), and enclosed by "C" and "S", as in

   "ChhS"

   The valid select digits for the 4001 are 0–15

   "C0S" – "C15S"

   Finally, the I/O options would be specified on a port-by-port basis with the connections to be made separated by commas, and enclosed in parentheses:

   "(n1, n2, n3 . . .)".

   where (n1, n2 . . .) are the option numbers associated with one I/O line. Hence, for a 4001 there will be four bracketed collections of I/O options.

   Each I/O pin has a series of 10 possible connections. These connections are consecutively numbered from 1-10. It is these numbers that should be in parentheses for each I/O pin.

   Example: "( )" indicates no connection
   "(1)" indicates only #1
   "(2,5,7)" indicates connections #2, 5 and 7.

   I/O options should be placed on the tape sequentially for the 4001 from I/O0–I/O3(4). Always avoid illegal combinations.

NOTE: Cards may also be submitted.

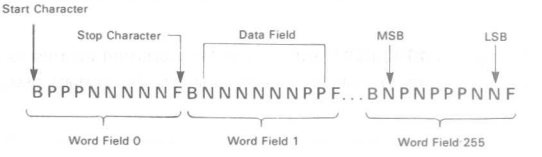
B. ROM Code

The format requirements are as follows:

1. All word fields are to be punched in consecutive order, starting with word field 0 (all addresses low). There must be exactly N word fields for the N x 8 ROM organization.

2. Each word field must begin with the start character B and end with the stop character F. There must be exactly 8 data characters between the B and F. Within the word field, a P results in a high level output (V_{SS} or logic 0 for MCS-40 CPUs) and a N results in a low level output (V_{DD} or logic 1 for MCS-40 CPUs).

   Example of 256 x 8 format (N=256):



3. Between word fields, comments not containing B's or F's may be inserted. Carriage return and line feed characters should be inserted (as a "comment") just before each word field (or at least between every four word fields). When these carriage returns, etc., are inserted, the tape may be easily listed on the teletype for purposes of error checking. It may be helpful to insert the word number (as a comment) at least every four word fields.

4. Within the ROM pattern words a character, "X", may be used. Where "P" and "N" indicate a "0" and "1" setting respectively, an "X" will indicate a single bit – "Don't Care" setting. This allows the optimum default bit values to be selected by Intel. The bit value will be fixed to allow for testing. The values will be specified to the user on the Verification Listing tape.

   In the place of a standard BPNF word, a "B*nF" word may be used. This indicates that the data in the last BPNF word encountered is to be repeated for the next n words ($1 \leq n \leq 1023$). Note that if a repeat count of 4 is given in word position 10, then words 10, 11, 12, and 13 will be repeats of word 9 (except for Don't Care bits which might conceivably have different assigned values).

   To indicate that an entire block (such as the remainder of a ROM) is not used (i.e., Don't Care), a word of Don't Care data can be followed by the remaining word count in a repeat count form.

MCS® CUSTOM ROM ORDER FORM

4001 ROM

CUSTOMER_____

P.O. NUMBER_____

DATE_____

For Intel use only

S#_____ PPPP_____

STD_____ ZZ_____

_____ DD_____

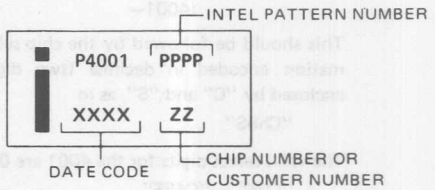
APP_____ DATE_____

All custom 4001 ROM orders must be submitted on this form. Programming information should be sent in the form of computer punched cards or punched paper tape per the formats designated on this order form. Additional forms are available from Intel.

MARKING

The marking as shown at the right must contain the Intel logo, the product type (P4001), the 4-digit Intel pattern number (PPPP), a date code (XXXX), and the 2-digit chip number (DD). An optional customer identification number may be substituted for the chip number (ZZ). Optional Customer Number (maximum 6 characters or spaces).

CUSTOMER NUMBER _____



MASK OPTION SPECIFICATIONS

A. CHIP NUMBER_____.
(Must be specified—any number from 0 through 15—DD).

B. I/O OPTION – Specify the connection numbers for each I/O pin (next page). Examples of some of the possible I/O options are shown below:

EXAMPLES – DESIRED OPTION/CONNECTIONS REQUIRED

1. Non-inverting output – 1 and 3 are connected.
2. Inverting output – 1 and 4 are connected.
3. Non-inverting input (no input resistor) – only 5 is connected.
4. Inverting input (input resistor to V_{SS}) – 2, 6, 7, and 9 are connected.
5. Non-inverting input (input resistor to V_{DD}) – 2, 7, 8, and 10 are connected.
6. If inputs and outputs are mixed on the same port, the pins used as the outputs must have the internal resistor connected to either V_{DD} or V_{SS} (8 and 9 or 8 and 10 must be connected). This is necessary for testing purposes. For example, if there are two inverting inputs (with no input resistor) and two non-inverting outputs, the connection would be made as follows:

   Inputs – 2 and 6 are connected
   Outputs – 1, 3, 8, and 9 are connected or
   1, 3, 8, and 10 are connected

If the pins on a port are all inputs or all outputs, the internal resistors do not have to be connected.

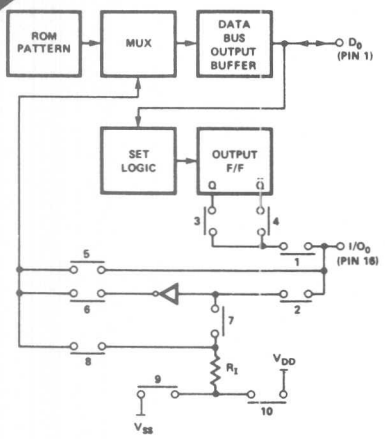
C. 4001 CUSTOM ROM PATTERN – Programming information should be sent in the form of computer punched cards or punched paper tape. In either case, a printout of the truth table must accompany the order. Based on the particular customer pattern, the characters should be written as a "P" for a high level output = V_{SS} (negative logic "0") or an "N" for a low level output = V_{DD} (negative logic "1").

Note that:

NOP = BPPPP PPPPF = 0000 0000

4001 I/O Options

SAMPLE

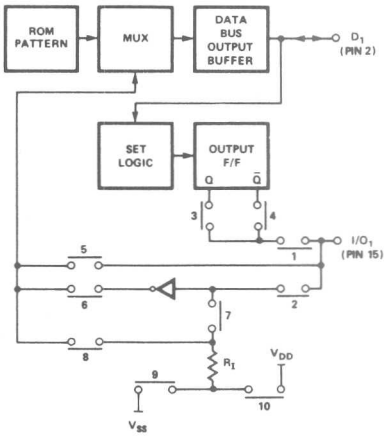


I/O_0 (PIN 16)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE CONNECTIONS ON SCHEMATIC)_____

a. For T^2L compatibility on the I/O lines the supply voltages should be $V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$

b. If non-inverting input option is used, $V_{IL} = -6.5$ Volts maximum (not TTL).

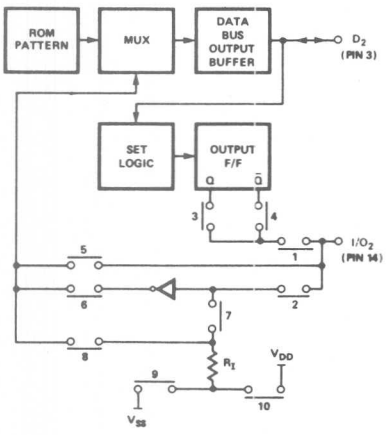


I/O_1 (PIN 15)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE CONNECTIONS ON SCHEMATIC)_____

a. For T^2L compatibility on the I/O lines the supply voltages should be $V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$

b. If non-inverting input option is used, $V_{IL} = -6.5$ Volts maximum (not TTL).

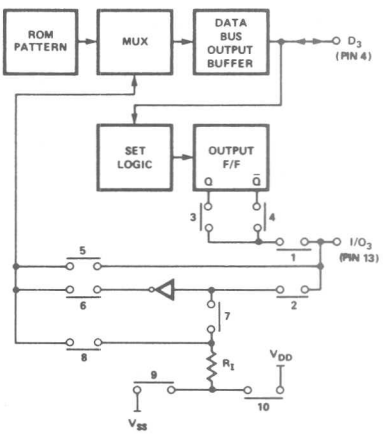


I/O_2 (PIN 14)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE CONNECTIONS ON SCHEMATIC)_____

a. For T^2L compatibility on the I/O lines the supply voltages should be $V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$

b. If non-inverting input option is used, $V_{IL} = -6.5$ Volts maximum (not TTL).



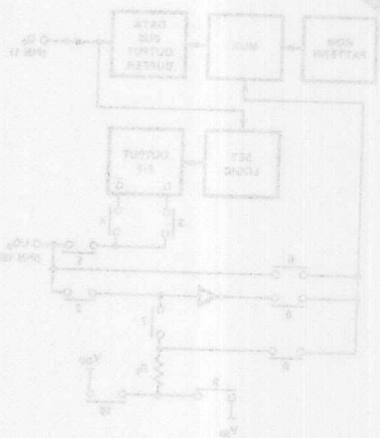
I/O_3 (PIN 13)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE CONNECTIONS ON SCHEMATIC)_____

a. For T^2L compatibility on the I/O lines the supply voltages should be $V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$

b. If non-inverting input option is used, $V_{IL} = -6.5$ Volts maximum (not TTL).

4001 I/O Options



I/O_0 (PIN 16)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE
CONNECTIONS ON SCHEMATIC)

a. For T_{ZL} compatibility on the I/O lines the supply voltage should be
$V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$
b. If non-inverting input option is used, $V_{IL} = +0.6$ Volts maximum (not TTL).

I/O_1 (PIN 15)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE
CONNECTIONS ON SCHEMATIC)

a. For T_{ZL} compatibility on the I/O lines the supply voltage should be
$V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$
b. If non-inverting input option is used, $V_{IL} = +0.6$ Volts maximum (not TTL).

I/O_2 (PIN 14)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE
CONNECTIONS ON SCHEMATIC)

a. For T_{ZL} compatibility on the I/O lines the supply voltage should be
$V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$
b. If non-inverting input option is used, $V_{IL} = +0.6$ Volts maximum (not TTL).

I/O_3 (PIN 13)

CONNECTIONS DESIRED (LIST NUMBERS & CIRCLE
CONNECTIONS ON SCHEMATIC)

a. For T_{ZL} compatibility on the I/O lines the supply voltage should be
$V_{DD} = -10V \pm 5\%$, $V_{SS} = +5V \pm 5\%$
b. If non-inverting input option is used, $V_{IL} = +0.6$ Volts maximum (not TTL).

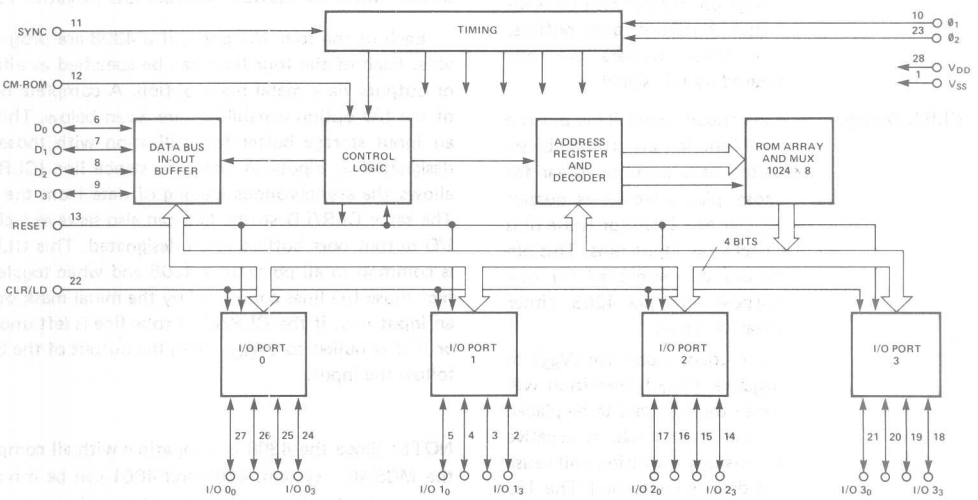
4308

1024 x 8 MASK PROGRAMMABLE ROM AND FOUR 4-BIT I/O PORTS

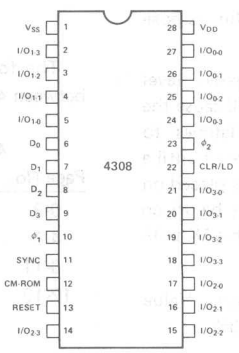
- Direct Interface to MCS-40™ 4-Bit Data Bus
- Equivalent to Four 4001 ROMs
- Four Independent 4-Bit I/O Ports
- Input I/O Buffer Storage with an Optional Strobe
- I/O Ports Low-Power TTL Compatible
- 28 Pin Dual In-Line Package
- Standard Operating Temperature Range of 0° to 70° C
- Also Available With -40° to +85° C Operating Range

The 4308 is a 1024 x 8 bit word ROM memory with four I/O ports. It is designed for the MCS-40™ system and is operationally compatible with all existing MCS-40 elements. The 4308 is functionally identical to four 4001 chips. The 4308 has 16 I/O lines arranged in four groups of four lines.

BLOCK DIAGRAM



PIN CONFIGURATION



Pin Description

Pin No.	Designation/ Type of Logic	Description of Function
1	V_{SS}	Most positive supply voltage.
2-5 14-17 18-21 24-27	$I/O1_3$-$I/O1_0$/Neg. $I/O2_3$-$I/O2_0$/Neg. $I/O3_3$-$I/O3_0$/Neg. $I/O0_3$-$I/O0_0$/Neg.	Four I/O ports consisting of 4 bidirectional and selectable lines.
6-9	D_0-D_3/Neg.	Bi-directional data bus. All information between processor and device is transmitted to these four pins.
10, 23	$\phi1$, $\phi2$/Neg.	Non-overlapped clock signals which determine device timing.
11	SYNC/Neg.	System synchronization signal generated by processor.
12	CM-ROM/Neg.	Chip enable generated by the processor.
13	RESET/Neg.	Reset input. A negative level (V_{DD}) on this pin will clear internal flip-flops and buffers. The input buffers are not cleared by this signal.
22	CLR/LD/Neg.	Clear/Load input. This pin is a dual function pin. It may be selected as a common Clear for those pins selected as output pins or as a Load for those pins selected as input pins. This pin should be designated for one purpose only per 4308, either Clear or Load. As a Load, a positive (V_{SS}) to negative (V_{DD}) transition will cause the I/O data to be placed in the input latch. A negative to positive transition will cause the data to be latched. The I/O pin state may be altered without changing the contents of the latch when the line is positive. As a Clear, a negative level (V_{DD}) on this line will cause the designated output latches to clear and remain cleared until a positive level (V_{SS}) is placed on the line. This line may be driven by a TTL output with a 1K pull-up resistor to V_{SS}.
28	V_{DD}	Main supply voltage. Value must be V_{SS} -15V ±5%.

Functional Description

The 4308 ROM program memory is arrayed 1024 x 8 bit words. For the program memory mode of operation, the A1 –A3 time periods of the instruction cycle are used to address the ROM contents. The 4308 decodes the first ten bits of the address to select 1 out of the 1024 words, 8 bits wide. The remaining two bits select a particular 4308, which has one of four possible metal option chip select addresses. Instruction information is available in two 4-bit segments during M_1 and M_2 time periods. A 4004 system can accommodate up to four 4308's while a 4040 system can utilize up to eight devices.

A second mode of operation of the ROM is as an Input/Output control device. In that mode a ROM chip will route information to and from data bus lines in and out of 4 I/O external lines. Each chip has the capability to identify itself for an I/O port operation, recognize an I/O port instruction and decide whether it is an Input or an Output operation and execute the instruction.

All internal flip flops (including the output register) will be reset when the RESET line goes low (negative voltage).

Each of the four I/O ports of a 4308 are program selectable. Each of the four lines can be specified as either inputs or outputs via a metal mask option. A complete description of the I/O option capabilities are given below. The 4308 has an input storage buffer for utilization with those I/O pins designated as inputs. A common strobe line (CLR/LD line) allows the asynchronous loading of data from the I/O lines. The same CLR/LD strobe line can also serve as a clear to the I/O output port buffers when designated. This CLR/LD line is common to all ports on a 4308 and when toggled, will effect those I/O lines connected by the metal mask option. For an input line, if the CLR/LD strobe line is left unconnected, or if it is pulled to (V_{DD}), then the output of the buffer will follow the input.

NOTE: Since the 4308 is compatible with all components of the MCS-40 system, 4308 and 4001 can be mixed on one memory bank as long as the chip select addresses are mutually exclusive.

The following table shows the chip number relationship between 4308 and 4001.

4308		4001	
Page No.	Chip No.	Page No.	Chip No.
0-3	(0)	0-15	0-15
4-7	(1)		
8-11	(2)		
12-15	(3)		

INSTRUCTION EXECUTION

The 4308 responds to the following instructions.

1. **SRC Instruction** (Send address to ROM and RAM)

   When the CPU executes an SRC instruction it will send out 8 bits of data during X_2 and X_3 and will activate the CM-ROM and one CM-RAM line at X_2. Data at X_2 (representing the contents of the first register of the register pair addressed by the SRC instruction), with simultaneous presence of CM-ROM, is interpreted by the 4308 as the chip number of the unit that should later perform an I/O operation. Data at X_3 is ignored. After an SRC only one CM-ROM and CM-RAM device will be selected.

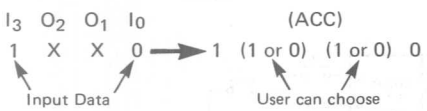
2. **WRR – Write ROM Port**

   The content of the accumulator is transferred to the ROM output port of the previously selected ROM chip. The data is available on the output pins until a new WRR is executed on the same chip. The ACC content and carry/link are unaffected. (The LSB bit of the accumulator appears on I/O_0.) No operation is performed on I/O lines coded as inputs.

3. **RDR – Read ROM Port**

   The data present at the input lines of the previously selected ROM chip is transferred to the accumulator.

   If the I/O option has both inputs and outputs within the same 4 I/O lines, the user can choose to have either "0" or "1" transferred to the accumulator for those I/O pins coded as outputs, when an RDR instruction is executed. For example, given a port with the I/O lines coded with 2 inputs and 2 outputs, when RDR is executed the transfer is as shown below:



Timing Considerations

At the beginning of each instruction sequence, a SYNC pulse is generated externally to synchronize the processor with the various components of the system. This pulse, along with the clock inputs ϕ_1 and ϕ_2, is used in the 4308 as an input to a timing register.

During time A_1, A_2, and A_3, the address is sequentially accepted from the data bus and decoded. During time A_3, the CM-ROM line will be active, and if the 2 highest order bits of the address sent at A_3 match the metal preprogrammed chip select option, the ROM will respond to the current address.

At time M_1 and M_2, the instruction OPR, OPA will be placed on the data bus for the processor.

After the SRC or Send Register Control instruction, which is used to designate a set of 4 I/O lines (1 port) on a particular ROM which are to be used for subsequent ROM I/O operations, is executed by the processor, the processor sends a 4 bit code to the ROM during X_2, and CM-ROM goes to a "1" (V_{DD}). The first two bits (D_3, D_2) of this code select a group of 1 out of 4 possible 4308, and the last two bits select a particular port (1 of 4 ports). This port remains selected until the next SRC instruction is executed.

In both the RDR and WRR operations, the CM-ROM line will become active during time M_2, and if the ROM has a previously selected I/O port, it will respond to the I/O in two ways. For a WRR accumulator, data will be transferred to an internal ROM selected output port flip-flops during X_2. Data will be available on the I/O line from time $X_3 \cdot \overline{\phi_2}$. The data will remain on the bus until a new WRR occurs, a reset occurs, or a clear (CLR/LD line) is generated. The RDR instruction will transfer information from the input port flip-flops of a previously selected port. Prior to RDR instruction, the user should insure that the input flip-flops have been loaded via the CLR/LD strobe if the load strobe is specified. If the load strobe is not specified, information on the input lines will be loaded into the accumulator at the time of the RDR.

I/O OPTIONS

The 4308 offers the following options on its I/O pins:

1. Input or output.
2. Inverted or direct (for input and output).
3. On-chip resistor connected to either V_{SS} or V_{DD} for input pins.
4. Asynchronous loading of input buffers via the CLR/LD signal.
5. Clear signal for any or all output ports via the CLR/LD signal.

Referring to the block diagram of the single I/O pin shown below which illustrates the various options available on a 4308, it should be noted that certain pin combinations are mutually exclusive and should not be specified together. There are also certain invalid combinations. The following combinations should be avoided:

8,9
5,6
3,4
10,11 – Both on a single pin and within a single 4308.

Examples of some common desired option/connections are:

a. *I/O pin inputs**

non-inverting	11, 2, 5, 7, 9 (TTL) – 2, 5, 7, 8
inverting	11, 2, 6, 7, 9 (TTL) – 2, 6, 7, 8

b. *I/O pin outputs*

non-inverting	3, 1 (10 optional)
inverting	4, 1 (10 optional)

Other combinations exist and should be used with caution.

**Option 11 need not be specified if an unbuffered input is desired. This is equivalent to a 4001 input.*

NOTE: The 4308 has the following enhancements over the 4001 as far as I/O options are concerned:

1. The capability of clearing any or all outputs with the CLR/LD signal.
2. TTL compatibility of both the inverting and non-inverting input paths for input ports.
3. The capability to select the LD option and have the input buffer become an input flip-flop and to have the CLR/LD signal become an asynchronous clock for loading data.

For TTL compatibility on the I/O lines, the supply voltage should be V_{DD} = -10V ±5%, V_{SS} = +5V ±5%. External pull-up is required for outputs.

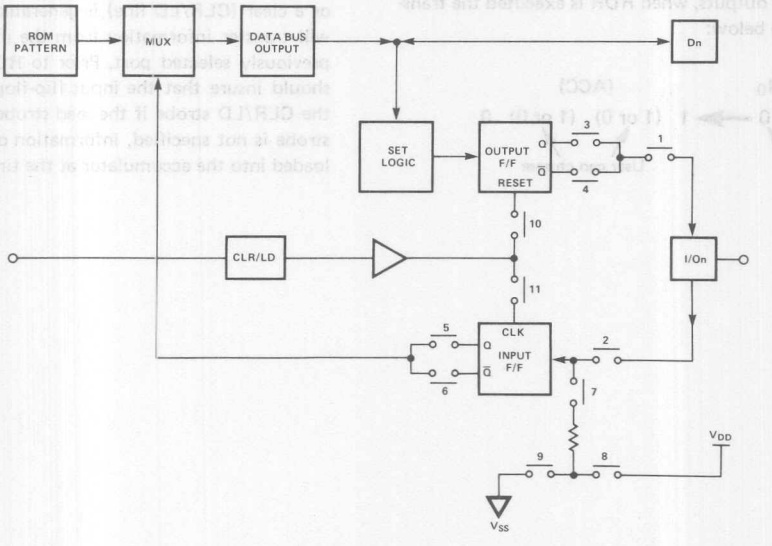


Figure 1. 4308 I/O Pin Options.

Absolute Maximum Ratings*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -55°C to + 125°C
Input Voltages and Supply Voltage
with respect to Vss ..... +0.5V to -20V
Power Dissipation ..... 1.0 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. and Operating Characteristics

T_A = 0°C to 70°C; V_{SS} -V_{DD} = 15V ±5%; $t_{\phi PW}$ = $t_{\phi D1}$ = 400 nsec; $t_{\phi D2}$ = 150 nsec; Logic "0" is defined as the more positive voltage (V_{IH}, V_{OH}); Logic "1" is defined as the more negative voltage (V_{IL}, V_{OL}); Unless Otherwise Specified.

SUPPLY CURRENT

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
I_{DD}	Average Supply Current		20	40	mA	T_A = 25°C
INPUT CHARACTERISTICS – ALL INPUTS EXCEPT I/O PINS						
I_{LI}	Input Leakage Current			10	μA	V_{IL} = V_{DD}
V_{IH}	Input High Voltage (Except Clocks)	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage (Except Clocks)	V_{DD}		V_{SS}-5.5	V	
V_{ILO}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	CLR/LD pin
V_{IHC}	Input High Voltage Clocks	V_{SS}-1.5		V_{SS}+.3	V	
V_{ILC}	Input Low Voltage Clocks	V_{DD}		V_{SS}-13.4	V	
OUTPUT CHARACTERISTICS – ALL OUTPUTS EXCEPT I/O PINS						
I_{LO}	Data Bus Output Leakage Current			10	μA	V_{OUT} = -12V
V_{OH}	Output High Voltage	V_{SS}-.5V	V_{SS}		V	Capacitive Load
I_{OL}	Data Lines Sinking Current	8	15		mA	V_{OUT} = V_{SS}
V_{OL}	Output Low Voltage, Data Bus, CM, SYNC	V_{SS}-12		V_{SS}-6.5	V	I_{OL} = 0.5mA
R_{OH}	Output Resistance, Data Line "0" Level		200	300	Ω	V_{OUT} = V_{SS} -.5V
I/O INPUT CHARACTERISTICS						
I_{LI}	Input Leakage Current			10	μA	
V_{IH}	Input High Voltage	V_{SS}-1.5		V_{SS}+.3	V	
V_{IL}	Input Low Voltage	V_{DD}		V_{SS}-4.2	V	
V_{IL}	CLR/LD Input Low Voltage	V_{DD}		V_{SS}-4.2	V	
R_I	Input Resistance, if Used	10	18	35	kΩ	R_I tied to V_{SS}; V_{IN} = V_{SS} -3V
R_I[1]	Input Resistance, if Used	15	25	40	kΩ	R_I tied to V_{DD}; V_{IN} = V_{SS} -3V
I/O OUTPUT CHARACTERISTICS						
V_{OH}	Output High Voltage	V_{SS}-.5V			V	I_{OUT} = 0
R_{OH}	I/O Output "0" Resistance		1.2	2	kΩ	V_{OUT} = V_{SS} -.5V
I_{OL}	I/O Output "1" Sink Current	2.5	5		mA	V_{OUT} = V_{SS} -.5V
I_{OL}[2]	I/O Output "1" Sink Current	0.8	3		mA	V_{OUT} = V_{SS} -4.85V
I_{CF}	I/O Output "1" Clamp Current			4	mA	V_{OUT} = V_{SS} -6V; T_A = 70°C
V_{OL}	I/O Output Low Voltage	V_{SS}-12		V_{SS}-6.5	V	I_{OUT} = 50μA

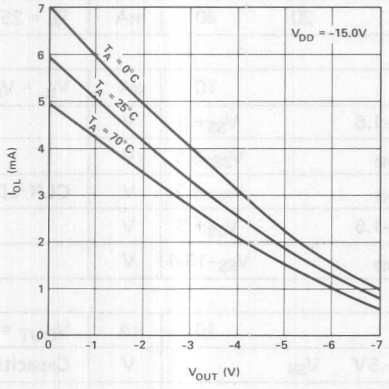
Notes: 1. R_I is large signal equivalent resistance to (V_{SS} -12) V.
2. For TTL compatability, use 12kΩ external resistor to V_{DD}.

D.C. and Operating Characteristics

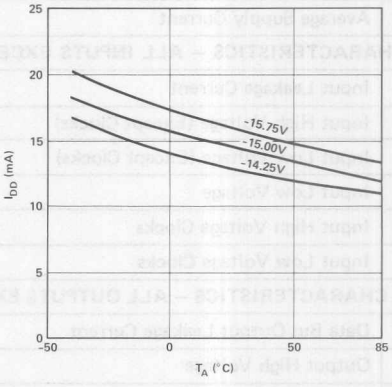
CAPACITANCE

Symbol	Parameter	Limit Min.	Typ.	Max.	Unit	Test Conditions
C_ϕ	Clock Capacitance		14	20	pF	$V_{IN} = V_{SS}$
C_{DB}	Data Bus Capacitance		7	10	pF	$V_{IN} = V_{SS}$
C_{IN}	Input Capacitance			10	pF	$V_{IN} = V_{SS}$
C_{OUT}	Output Capacitance			10	pF	$V_{IN} = V_{SS}$

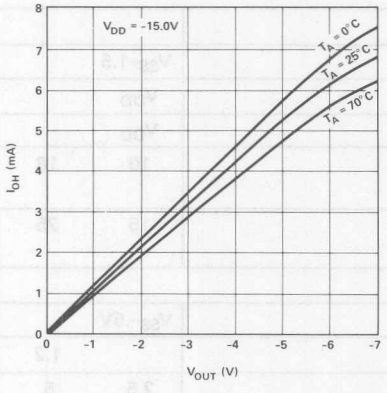
4308 OUTPUT PINS ("1" LEVEL)



4308 SUPPLY CURRENT VS. TEMPERATURE



4308 OUTPUT PINS ("0" LEVEL)



A.C. Characteristics

$T_A = 0°C$ to $70°C$, $V_{SS} - V_{DD} = 15V \pm 5\%$

Symbol	Parameter	Limit Min.	Limit Typ.	Limit Max.	Unit	Test Conditions
t_{CY}	Clock Period	1.35		2.0	μsec	
$t\phi_R$	Clock Rise Time			50	ns	
$t\phi_F$	Clock Fall Time			50	ns	
$t\phi_{PW}$	Clock Width	380		480	ns	
$t\phi_{D1}$	Clock Delay ϕ_1 to ϕ_2	400		550	ns	
$t\phi_{D2}$	Clock Delay ϕ_2 to ϕ_1	150			ns	
t_W	Data-In, CM, SYNC Write Time	350	100		ns	
t_H[1,3]	Data-In, CM, SYNC Hold Time	40	20		ns	
t_{OS}[2]	Set Time (Reference)	0			ns	
t_{ACC}	Data-Out Access Time					C_{OUT} =
	Data Lines			930	ns	500pF Data Lines[4]
	SYNC			930	ns	500pF SYNC
	CM-ROM			930	ns	160pF CM-ROM
	CM-RAM			930	ns	50pF CM-RAM
t_{OH}	Data-Out Hold Time	50	150		ns	C_{OUT} = 20pF
t_{IS}	I/O Input Set-Time	50			ns	
t_{IH}	I/O Input Hold-Time	100			ns	
$t_{PW\ I/O}$	C/L Pulse-Width	1000	400		ns	
$t_{W\ C/L}$	C/L Write Time	350	200		ns	
$t_{H\ C/L}$	C/L Hold Time	100			ns	
t_D	I/O Output Delay			1500	ns	C_{OUT} = 100pF
t_C[5]	I/O Output Delay on C/L		750	1500	ns	C_{OUT} = 100pF
$t_{W\ \phi 2F}$[6]	Data In Write Time with Respect to ϕ_2	-30	-60		ns	

Notes:
1. t_H measured with $t_{\phi R}$ = 10nsec.
2. T_{ACC} is Data Bus, SYNC and CM-line output access time referred to the ϕ_2 trailing edge which clocks these lines out. t_{OS} is the same output access time referred to the leading edge of the next ϕ_2 clock pulse.
3. All MCS-40 components which may transmit instruction or data to 4004/4040 at M_2 and X_2 always enter a float state until the 4004/4040 takes over the data bus at X_1 and X_3 time. Therefore the t_H requirement is always insured since each component contributes 10μA of leakage current and 10pF of capacitance which guarantees that the data bus cannot change faster than 1V/μs.
4. t_{ACC}, 4308 is guaranteed with $t_{\phi D2}$ = 200 nsec.
5. C/L Clears output buffer when low. C/L enters data into input buffer when low. C/L rising edge latches input buffer. Port Option 10 and 11 are mutually exclusive on any 4308.
6. Data Bus Inputs are guaranteed valid before ϕ_2 falling edge by 4004, 4040 t_{ACC}. If $t_{PW\phi 2}$ is widened, then t_{CY} is increased and Data Bus Inputs remain valid before ϕ_2 falling edge. Thus, $t_{W\phi 2F}$ is not a system constraint.

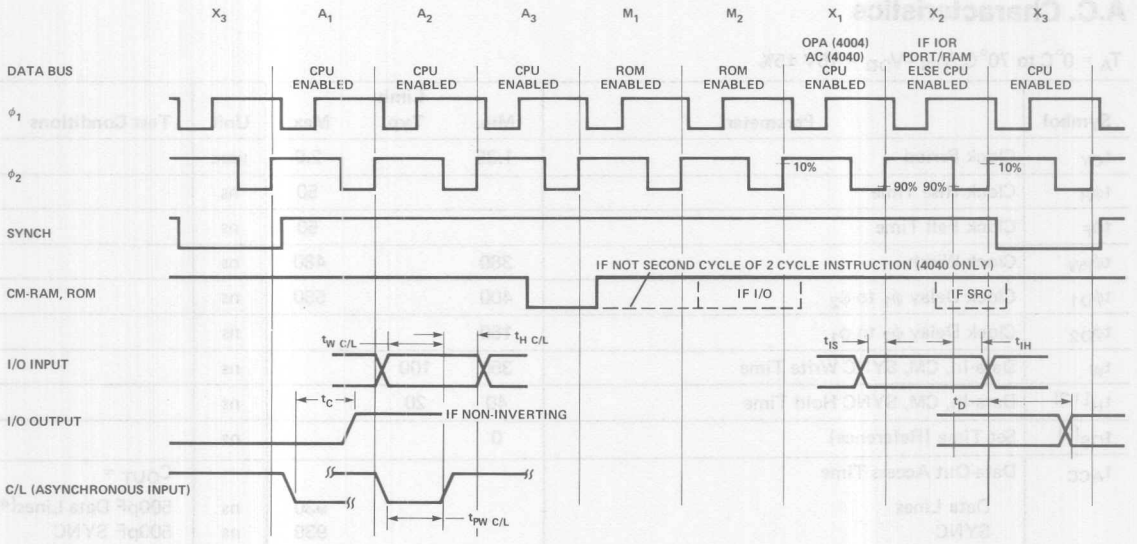


Figure 2. Timing Diagram.

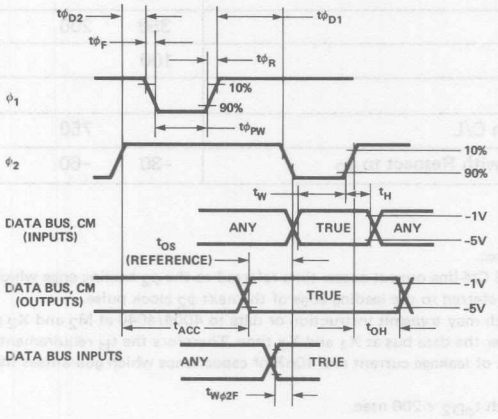


Figure 3. Timing Detail.

Programming Instruction

To insure optimum handling of ROM programs and avoid delays, programs should be specified in the following format.

Paper Tape Format*

A 1" wide paper tape using 8 bit ASCII code, such as a model 33ASR teletype produces:

A. Preamble

1. Preceding the first word field and following the last word field, there should be a leader/trailer length of at least 25 characters. This should consist of rubout punches.

2. Included in the tape before the leader, and preceded by another leader, should be the customer's complete telex or twx number and if more than one pattern is being transmitted, the ROM pattern number.

3. The first ROM pattern preamble field is the device type number or ROM number. The field should be framed by an "I" and "–"

   I4308–

   This should be followed by the chip select information encoded in decimal (two digits), and enclosed by "C" and "S", as in

   "ChhS"

   The valid select digits for the 4308 are 0–3

   "C0S" – "C3S"

   Finally, the I/O options would be specified on a port-by-port basis with the connections to be made separated by commas, and enclosed in parentheses:

   "(n1, n2, n3 . . .)".

   where (n1, n2 . . .) are the option numbers associated with one I/O line. Hence, for the 4308 there will be sixteen bracketed collections of I/O options.

   Each I/O pin has a series of 11 possible connections. These connections are consecutively numbered from 1-11. It is these numbers that should be in parentheses for each I/O pin.

   Example: "( )" indicates no connection
   "(1)" indicates only #1
   "(2,5,7)" indicates connections #2, 5 and 7.

   I/O options should be placed on the tape sequentially for the 4308, from $I/O0_0$ – $I/O3_3$(16). Always avoid illegal combinations.

NOTE: Cards may also be submitted.

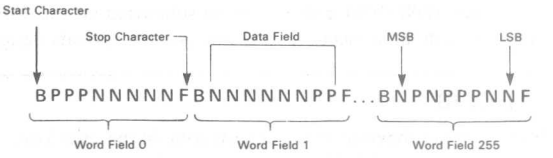
B. ROM Code

The format requirements are as follows:

1. All word fields are to be punched in consecutive order, starting with word field 0 (all addresses low). There must be exactly N word fields for the N x 8 ROM organization.

2. Each word field must begin with the start character B and end with the stop character F. There must be exactly 8 data characters between the B and F. Within the word field, a P results in a high level output (V_{SS} or logic 0 for MCS-40 CPUs) and a N results in a low level output (V_{DD} or logic 1 for MCS-40 CPUs).

   Example of 256 x 8 format (N=256):



3. Between word fields, comments not containing B's or F's may be inserted. Carriage return and line feed characters should be inserted (as a "comment") just before each word field (or at least between every four word fields). When these carriage returns, etc., are inserted, the tape may be easily listed on the teletype for purposes of error checking. It may be helpful to insert the word number (as a comment) at least every four word fields.

4. Within the ROM pattern words a character, "X", may be used. Where "P" and "N" indicate a "0" and "1" setting respectively, an "X" will indicate a single bit – "Don't Care" setting. This allows the optimum default bit values to be selected by Intel. The bit value will be fixed to allow for testing. The values will be specified to the user on the Verification Listing tape.

   In the place of a standard BPNF word, a "B*nF" word may be used. This indicates that the data in the last BPNF word encountered is to be repeated for the next n words ($1 \leq n \leq 1023$). Note that if a repeat count of 4 is given in word position 10, then words 10, 11, 12, and 13 will be repeats of word 9 (except for Don't Care bits which might conceivably have different assigned values).

   To indicate that an entire block (such as the remainder of a ROM) is not used (i.e., Don't Care), a word of Don't Care data can be followed by the remaining word count in a repeat count form.

MCS® CUSTOM ROM ORDER FORM

4308 ROM

CUSTOMER _____

P.O. NUMBER _____

DATE _____

For Intel use only

S# _____ PPPP _____

STD _____ ZZ _____

_____ DD _____

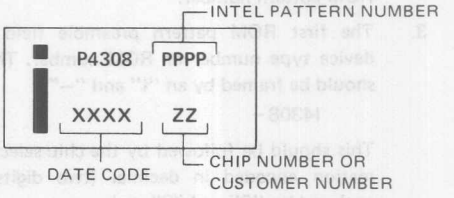
APP _____ DATE _____

All custom 4308 ROM orders must be submitted on this form. Programming information should be sent in the form of computer punched cards or punched paper tape per the formats designated on this order form. Additional forms are available from Intel.

MARKING

The marking as shown at the right must contain the Intel logo, the product type (P4308), the 4-digit Intel pattern number (PPPP), a date code (XXXX), and the 2-digit chip number (DD). An optional customer identification number may be substituted for the chip number (ZZ). Optional Customer Number (maximum 6 characters or spaces).

CUSTOMER NUMBER _____



MASK OPTION SPECIFICATION

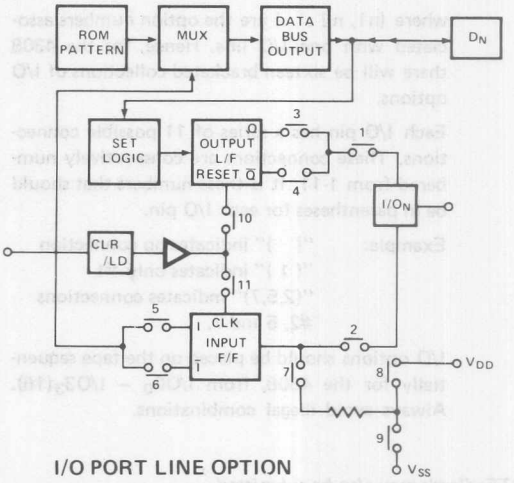
A. CHIP NUMBER _____ (Must be specified).

B. I/O OPTION — Specify the connection numbers for each I/O pin. See table below.

C. 4308 CUSTOM ROM PATTERN — Programming information should be sent in the form of computer punched cards or punched paper tape. In either case, a printout of the truth table must accompany the order. Based on the particular customer pattern, the characters should be written as a "P" for a high level output = V_{SS} (negative logic "0") or an "N" for a low level output = V_{DD} (negative logic "1").

Note that: NOP = BPPPP PPPPF = 0000 0000

PIN		OPTION										
I/O 0_0	27	1	2	3	4	5	6	7	8	9	10	11
I/O 0_1	26	1	2	3	4	5	6	7	8	9	10	11
I/O 0_2	25	1	2	3	4	5	6	7	8	9	10	11
I/O 0_3	24	1	2	3	4	5	6	7	8	9	10	11
I/O 1_0	5	1	2	3	4	5	6	7	8	9	10	11
I/O 1_1	4	1	2	3	4	5	6	7	8	9	10	11
I/O 1_2	3	1	2	3	4	5	6	7	8	9	10	11
I/O 1_3	2	1	2	3	4	5	6	7	8	9	10	11
I/O 2_0	17	1	2	3	4	5	6	7	8	9	10	11
I/O 2_1	16	1	2	3	4	5	6	7	8	9	10	11
I/O 2_2	15	1	2	3	4	5	6	7	8	9	10	11
I/O 2_3	14	1	2	3	4	5	6	7	8	9	10	11
I/O 3_0	21	1	2	3	4	5	6	7	8	9	10	11
I/O 3_1	20	1	2	3	4	5	6	7	8	9	10	11
I/O 3_2	19	1	2	3	4	5	6	7	8	9	10	11
I/O 3_3	18	1	2	3	4	5	6	7	8	9	10	11



I/O PORT LINE OPTION

NOTE: Options 10 and 11 cannot both be specified.

4316A

16,384 BIT STATIC MOS READ ONLY MEMORY

Organization: 2048 Words x 8 Bits
Access Time: 850 ns Max.

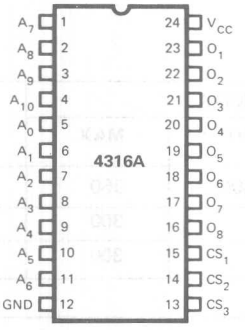
- **Single +5 Volts Power Supply Voltage**
- **Directly TTL Compatible — All Inputs and Outputs**
- **Low Power Dissipation of 31.4 μW/Bit Maximum**
- **Three-State Output — OR-Tie Capability**
- **Fully Decoded — On Chip Address Decode**
- **Interface to 4004/4040 CPU Via 4008/4009 or 4289 Standard Memory Interface**
- **Standard Operating Temperature Range of 0° to 70°C**
- **Also Available with -40° to +85°C Operating Range**

The Intel® 4316A is a 16,384-bit static MOS read only memory organized as 2048 words by 8 bits. This ROM is designed for microcomputer memory applications where high performance, large bit storage, and simple interfacing are important design objectives. It can be used in MCS-40™ systems via the 4008/4009 or 4289 Standard Memory Interface components.

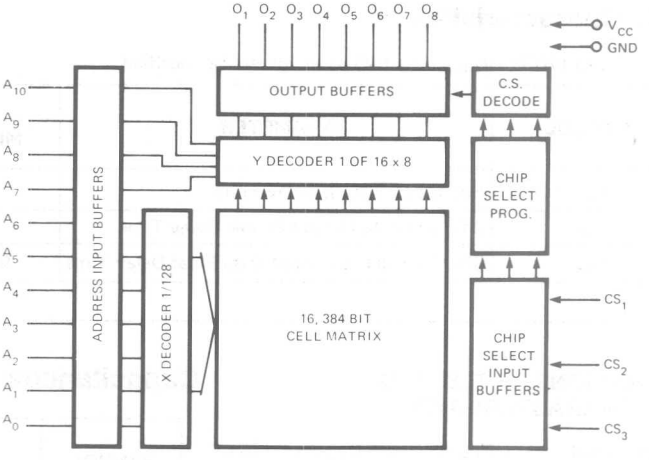
The inputs and outputs are fully TTL compatible. This device operates with a single +5V power supply. The three chip select inputs are programmable. Any combination of active high or low level chip select inputs can be defined and the desired chip select code is fixed during the masking process. These three programmable chip select inputs, as well as OR-tie compatibility on the outputs, facilitate easy memory expansion.

The 4316A read only memory is fabricated with N-channel silicon gate technology. This technology provides the designer with high performance, easy-to-use MOS circuits. Only a single +5V power supply is needed and all devices are directly TTL compatible.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

A_0-A_{10}	ADDRESS INPUTS
O_1-O_8	DATA OUTPUTS
CS_1-CS_3	PROGRAMMABLE CHIP SELECT INPUTS

Absolute Maximum Ratings*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage On Any Pin With Respect To Ground . . . . . −0.5V to +7V
Power Dissipation . . . . . 1.0 Watt

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. and Operating Characteristics

T_A = 0°C to +70°C, V_{CC} = 5V ±5% unless otherwise specified

SYMBOL	PARAMETER	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.(1)	MAX.		
I_{LI}	Input Load Current (All Input Pins)			10	μA	V_{IN} = 0 to 5.25V
I_{LOH}	Output Leakage Current			10	μA	CS = 2.2V, V_{OUT} = 4.0V
I_{LOL}	Output Leakage Current			−20	μA	CS = 2.2V, V_{OUT} = 0.45V
I_{CC}	Power Supply Current		40	98	mA	All inputs 5.25V Data Out Open
V_{IL}	Input "Low" Voltage	−0.5		0.8	V	
V_{IH}	Input "High" Voltage	2.0		V_{CC}+1.0V	V	
V_{OL}	Output "Low" Voltage			0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output "High" Voltage	2.2			V	I_{OH} = −100 μA

(1) Typical values for T_A = 25°C and nominal supply voltage.

A.C. Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±5% unless otherwise specified

SYMBOL	PARAMETER	LIMITS			UNIT
		MIN.	TYP.(1)	MAX.	
t_A	Address to Output Delay Time		400	850	nS
t_{CO}	Chip Select to Output Enable Delay Time			300	nS
t_{DF}	Chip Deselect to Output Data Float Delay Time	0		300	nS

CONDITIONS OF TEST FOR A.C. CHARACTERISTICS

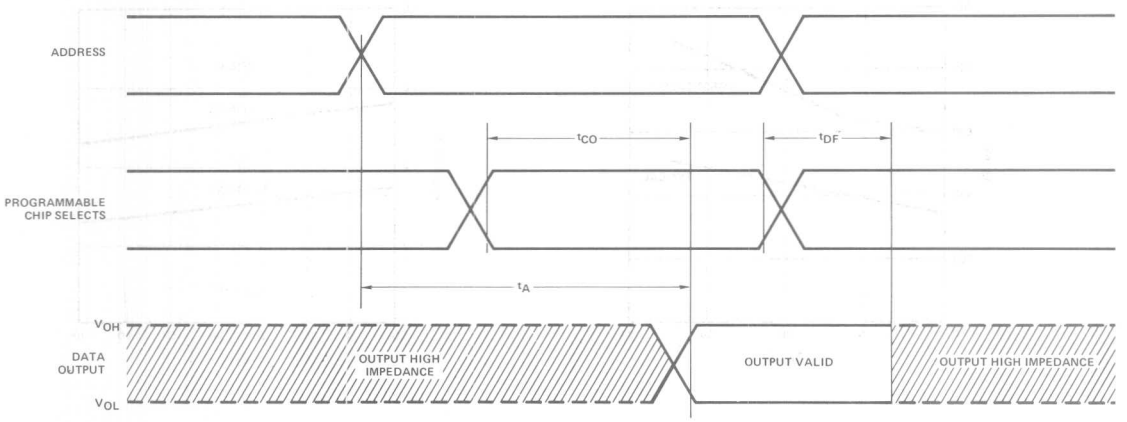
Output Load . . . 1 TTL Gate, and C_{LOAD} = 100 pF
Input Pulse Levels . . . . . 0.8 to 2.0V
Input Pulse Rise and Fall Times . (10% to 90%) 20 nS
Timing Measurement Reference Level
Input 1.5V
Output . . . . . 0.45V to 2.2V

Capacitance(2) T_A = 25°C, f = 1 MHz

SYMBOL	TEST	LIMITS	
		TYP.	MAX.
C_{IN}	All Pins Except Pin Under Test Tied to AC Ground	4 pF	10 pF
C_{OUT}	All Pins Except Pin Under Test Tied to AC Ground	8 pF	15 pF

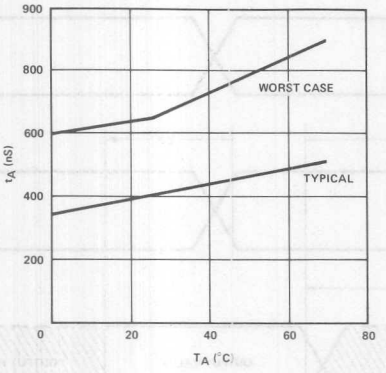
(2) This parameter is periodically sampled and is not 100% tested.

Waveforms

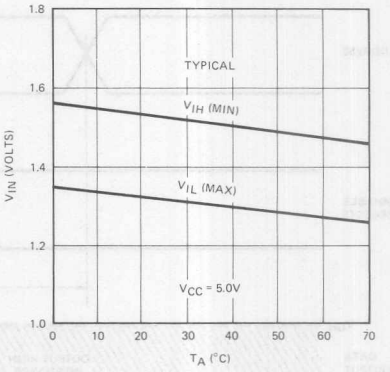

ADDRESS
t_{CO}
t_{DF}
PROGRAMMABLE CHIP SELECTS
t_A
V_{OH}
DATA OUTPUT
V_{OL}
OUTPUT HIGH IMPEDANCE
OUTPUT VALID
OUTPUT HIGH IMPEDANCE

Typical D.C. Characteristics

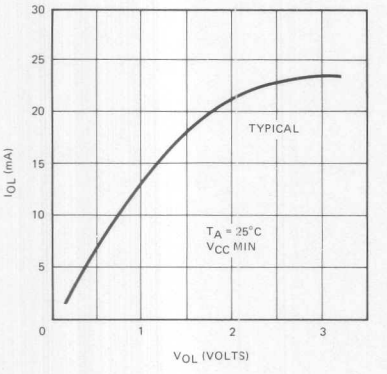
ACCESS TIME VS. AMBIENT TEMPERATURE



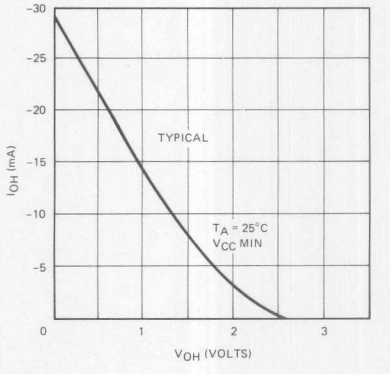
V_{IN} LIMITS VS. TEMPERATURE



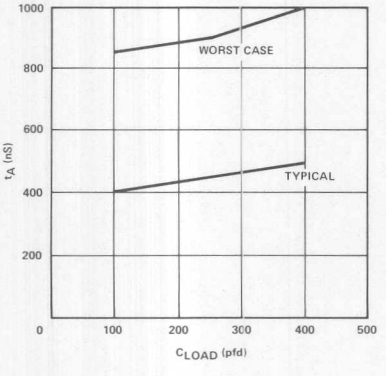
OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



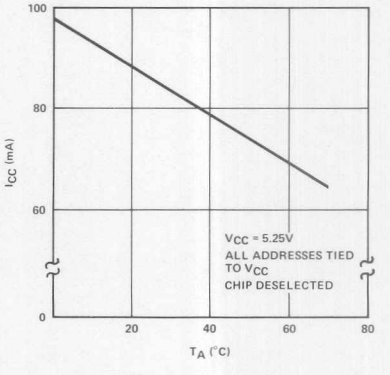
OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



ACCESS TIME VS. LOAD CAPACITANCE



STATIC I_{CC} VS. AMBIENT TEMPERATURE WORST CASE



MCS® CUSTOM ROM ORDER FORM

4316A ROM

CUSTOMER _____

P.O. NUMBER _____

DATE _____

For Intel use only

S# _____ PPPP _____

STD _____ ZZ _____

_____ DD _____

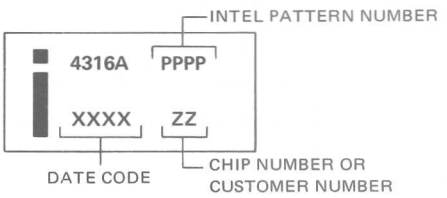
APP _____ DATE _____

All custom 4316A ROM orders must be submitted on this form. Programming information should be sent in the form of computer punched cards or punched paper tape per the formats designated on this order form. Additional forms are available from Intel.

MARKING

The marking as shown at the right must contain the Intel® logo, the product type (P4316A), the 4-digit Intel pattern number (PPPP), a date code (XXXX), and the 2-digit chip number (DD). An optional customer identification number may be substituted for the chip number (ZZ). Optional Customer Number (maximum 9 characters or spaces).

CUSTOMER NUMBER _____



MASK OPTION SPECIFICATIONS

A. CHIP NUMBER _____
(Must be specified—any number from 0 through 7—DD).

The chip number will be coded in terms of positive logic where a logic "1" is a high level input.

Chip Number	CS3	CS2	CS1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

B. ROM Truth Table Format

Programming information should be sent in the form of computer punched cards or punched paper tape. In either case, a printout of the truth table should be accompanied with the order.

The following general format is applicable to the programming information sent to Intel:

- Data fields should be ordered beginning with the least significant address (0000) and ending with the most significant address (2047).
- A data field should start with the most significant bit and end with the least significant bit.
- The data field should consist of P's and N's. A P is to indicate a high level output (most positive) and an N a low level output (most negative). In terms of positive logic, a P is defined as a logic "1" and an N is defined as a logic "0".

NOTE:

Memory address, memory data, I/O bus, and chip select lines from 4289 or a 4008/4009 are defined with respect to positive logic. The MCS-40 data and control lines from the CPU are defined with respect to negative logic. As a result, in 4316A program memory used with the 4289 or 4009 programs should be coded with logic "1" = high level and logic "0" = low level (i.e., NOP = 0000 0000 = NNNN NNNN).

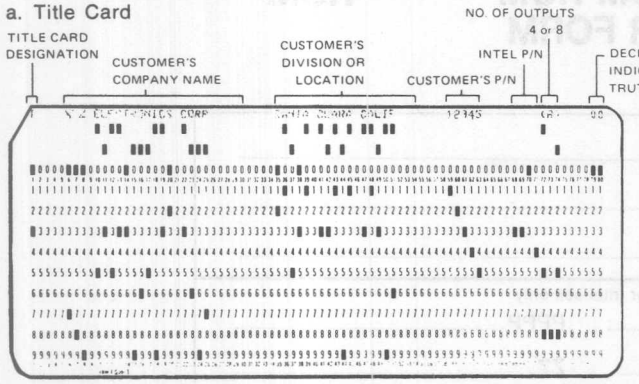
If the programming information is sent on a punched paper tape, then a start character, B, and an end character, F, must be used in the data field.

1. Punched Card Format

An 80-column Hollerith card (preferably interpreted) punched by an IBM 026 or 029 keypunch should be submitted. The first card will be a title card; the format is as follows:

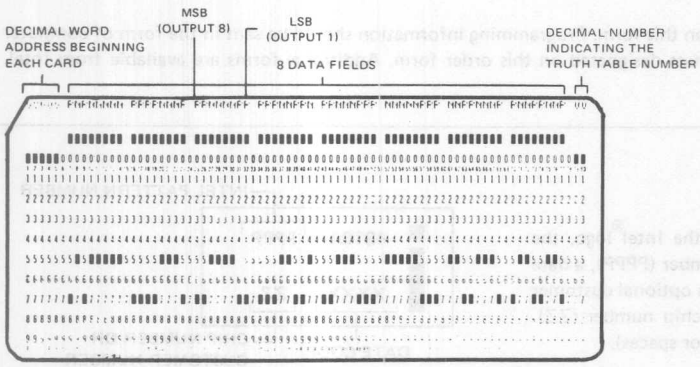
MCS® CUSTOM ROM ORDER FORM

a. Title Card



Column	Data
1	Punch a T
2–5	Blank
6–30	Customer Company Name
31–34	Blank
35–54	Customer's Company Division or location
55–57	Blank
58–66	Customer Part Number
67	Blank
68–75	Punch the Intel 4-digit basic part number and in ( ) the number of output bits, e.g., 8316A(8).
76–78	Blank
79–80	Punch a 2-digit decimal number to identify the truth table number (mask programmed chip select number).

b. For a 2048 word X 8-bit organization only, cards 2 and the following cards should be punched as shown.



Column	Data
1–5	Punch the 5-digit decimal equivalent of the binary coded location which begins each card. The address is right justified, i.e., 00000, 00008, 00016, etc.
6	Blank
7–14	Data Field
15	Blank
16–23	Data Field
33	Blank
34–41	Data Field
42	Blank
43–50	Data Field
51	Blank
52–59	Data Field
60	Blank
61–68	Data Field
69	Blank
70–77	Data Field
78	Blank
79–80	Punch same 2-digit decimal number as in title card.

2. Paper Tape Format

1" wide paper tape using 7- or 8-bit ASCII code, such as a model 33 ASR teletype produces, or the 11/16" wide paper tape using a 5-bit Baudot code, such as a Telex produces.

The format requirements are as follows:

a. All word fields are to be punched in consecutive order, starting with word field 0 (all addresses low). There must be exactly 2048 word fields for the 2048 X 8 ROM organization.

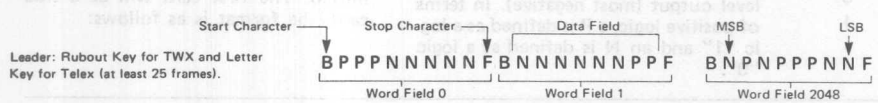
b. Each word field must begin with the start character B and end with the stop character F. There must be exactly 8 data characters between the B and F. NO OTHER CHARACTERS, SUCH AS RUBOUTS, ARE ALLOWED ANYWHERE IN A WORD FIELD. If in preparing a tape an error is made, the entire word field, including the B and F, must be rubbed out. Within the word field, a P results in a high level output and an N results in a low level output.

c. Preceding the first word field and following the last word field, there must be a leader/trailer length of *at least 25 characters. This should consist of rubout or null punches (letter key for Telex tapes).*

d. Between word fields, comments not containing B's or F's may be inserted. Carriage return and line feed characters should be inserted as a "comment") just before each word field (or at least between every four word fields). When these carriage returns, etc., are inserted, the tape may be easily listed on the teletype for purposes of error checking. The customer may also find it helpful to insert the word number (as a comment) at least every four word fields.

e. Included in the tape before the leader should be the customer's complete Telex or TWX number and, if more than one pattern is being transmitted, the ROM pattern number.

f. MSB and LSB are the most and least significant bit of the device outputs. Refer to the data sheet for the pin numbers.



REPROGRAMMABLE 2K PROM

- **Access Time: 1.7 usec Max.**
- **Fast Programming: 2 Minutes for all 2048 Bits**
- **Ultraviolet Erasable and Electronically Reprogrammable**
- **Fully Decoded, 256 x 8 Organization**
- **Static MOS: No Clocks Required**
- **Inputs and Outputs TTL Compatible**
- **Three-State Output: OR-Tie Capability**
- **Standard Operating Temperature Range of 0° to 70° C**
- **Also Available with -40° to +85° C Operating Range**

The 4702A is a 256 word by 8 bit electrically programmable ROM ideally suited for microcomputer system development where fast turn-around and pattern experimentation are important. The 4702A undergoes complete programming and functional testing on each bit position prior to shipment, thus insuring 100% programmability.

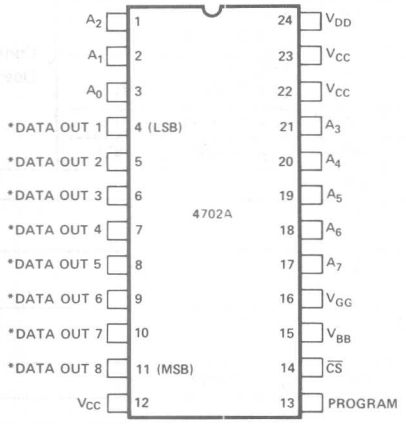
The 4702A is packaged in a 24 pin dual-in-line package with a transparent quartz lid. The transparent quartz lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device. This procedure can be repeated as many times as required.

The circuitry of the 4702A is entirely static; no clocks are required.

A pin-for pin metal mask programmed ROM, the Intel® 1302A, is ideal for large volume production runs of systems initially using the 4702A.

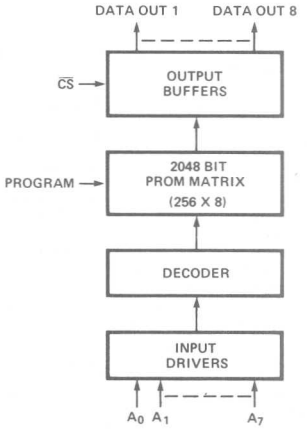
The 4702A is fabricated with silicon gate technology. This low threshold technology allows the design and production of higher performance MOS circuits and provides a higher functional density on a monolithic chip than conventional MOS technologies.

PIN CONFIGURATION



*THIS PIN IS THE DATA INPUT LEAD DURING PROGRAMMING.

BLOCK DIAGRAM



PIN NAMES

A_0-A_7	ADDRESS INPUTS
$\overline{CS}$	CHIP SELECT INPUT
DO_1- DO_2	DATA OUTPUTS

Pin Connections

The external lead connections to the **4702A** differ, depending on whether the device is being programmed[1] or used in read mode. (See following table.)

MODE \ PIN	12 (V_{CC})	13 (Program)	14 ($\overline{CS}$)	15 (V_{BB})	16 (V_{GG})	22 (V_{CC})	23 (V_{CC})
Read	V_{CC}	V_{CC}	GND	V_{CC}	V_{GG}	V_{CC}	V_{CC}
Programming	GND	Program Pulse	GND	V_{BB}	Pulsed V_{GG} (V_{IL4P})	GND	GND

Absolute Maximum Ratings*

Ambient Temperature Under Bias . . . . . 0°C to +70°C
Storage Temperature . . . . . −65°C to +125°C
Soldering Temperature of Leads (10 sec) . . . . . +300°C
Power Dissipation . . . . . 2 Watts
Read Operation: Input Voltages and Supply Voltages with respect to V_{CC} . . . . . +0.5V to −20V
Program Operation: Input Voltages and Supply Voltages with respect to V_{CC} . . . . . −48V

*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.

READ OPERATION

D.C. and Operating Characteristics

T_A = 0°C to 70°C, V_{CC} = +5V±5%, V_{DD} = -10V±5%, V_{GG} = -10V±5%, unless otherwise noted.

SYMBOL	TEST	MIN.	TYP.[2]	MAX.	UNIT	CONDITIONS
I_{LI}	Address and Chip Select Input Load Current			10	μA	V_{IN} = 0.0V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 0.0V, $\overline{CS}$ = V_{CC} −2
I_{DD1}	Power Supply Current		39	54	mA	$\overline{CS}$=V_{CC} −2, I_{OL} =0.0mA, T_A = 25°C
I_{DD2}	Power Supply Current		36	50	mA	$\overline{CS}$=0.0, I_{OL} =0.0mA, T_A = 25°C
I_{DD3}	Power Supply Current		43	63	mA	$\overline{CS}$=V_{CC} −2, I_{OL} =0.0mA, T_A = 0°C
I_{CF1}	Output Clamp Current		8	14	mA	V_{OUT} = −1.0V, T_A = 0°C
I_{CF2}	Output Clamp Current			13	mA	V_{OUT} = −1.0V, T_A = 25°C
						} Continuous Operation
I_{GG}	Gate Supply Current			10	μA	
V_{IL1}	Input Low Voltage for TTL Interface	−1.0		0.65	V	
V_{IL2}	Input Low Voltage for MOS Interface	V_{DD}		V_{CC} −6	V	
V_{IH}	Address and Chip Select Input High Voltage	V_{CC} −2		V_{CC} +0.3	V	
I_{OL}	Output Sink Current	1.6	4		mA	V_{OUT} = 0.45V
V_{OL}	Output Low Voltage		.7	0.45	V	I_{OL} = 1.6mA
V_{OH}	Output High Voltage	3.5			V	I_{OH} = −100 μA

Note 1: In the programming mode, the data inputs 1−8 are pins 4−11 respectively. $\overline{CS}$ = GND.
Note 2: Typical values are at nominal voltages and T_A = 25°C.

A.C. Characteristics

T_A = 0° C to +70°C, V_{CC} = +5V ±5%, V_{DD} = -10V ±5%, V_{GG} = -10V ±5% unless otherwise noted

SYMBOL	TEST	MINIMUM	TYPICAL	MAXIMUM	UNIT
Freq.	Repetition Rate			1	MHz
t_{OH}	Previous read data valid			100	ns
t_{ACC}	Address to output delay			1.7	μs
t_{CS}	Chip select delay			800	ns
t_{CO}	Output delay from $\overline{CS}$			900	ns
t_{OD}	Output deselect			300	ns

Capacitance* T_A = 25°C

SYMBOL	TEST	MINIMUM	TYPICAL	MAXIMUM	UNIT	CONDITIONS
C_{IN}	Input Capacitance		8	15	pF	$V_{IN}=V_{CC}$, $CS=V_{CC}$, $V_{OUT}=V_{CC}$, $V_{GG}=V_{CC}$ — All unused pins are at A.C. ground
C_{OUT}	Output Capacitance		10	15	pF	

*This parameter is periodically sampled and is not 100% tested.

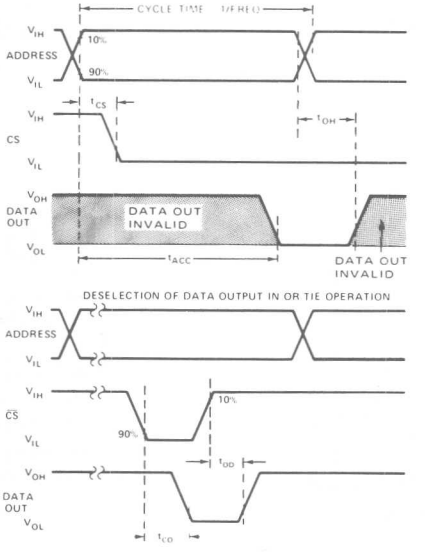
Switching Characteristics

Conditions of Test:

Input pulse amplitudes: 0 to 4V; t_R, t_F ≤50 ns.

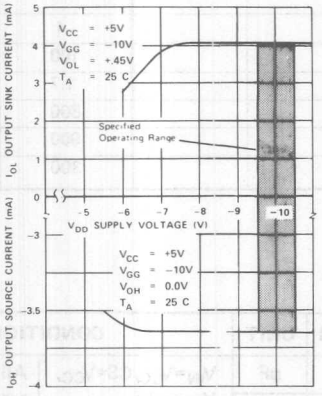
a) For output load = 1 TTL gate; measurements made at output of TTL gate (t_{PD} ≤15 ns)

b) For pure capacitive load of 75pf.

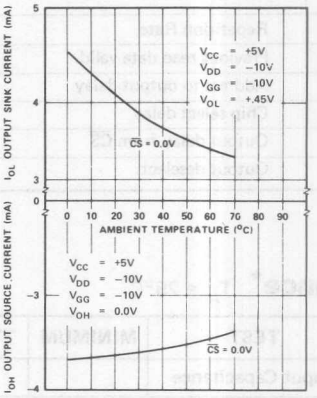


Typical Characteristics

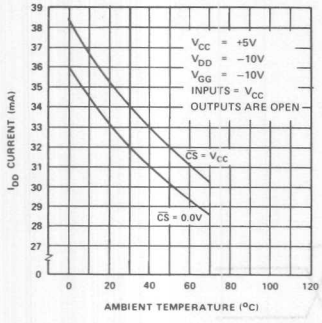
OUTPUT CURRENT VS. V_{DD} SUPPLY VOLTAGE



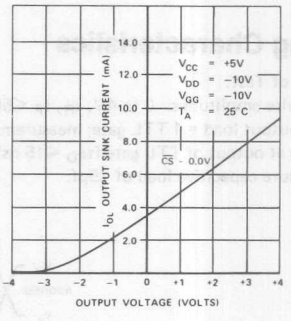
OUTPUT CURRENT VS. TEMPERATURE



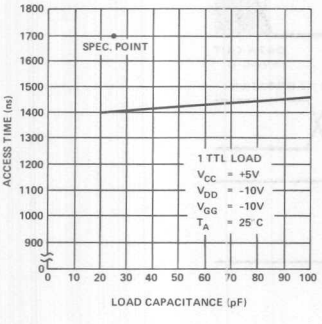
I_{DD} CURRENT VS. TEMPERATURE



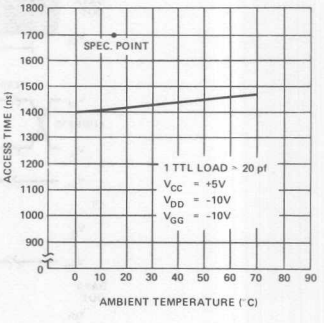
OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



ACCESS TIME VS. LOAD CAPACITANCE



ACCESS TIME VS. TEMPERATURE



PROGRAMMING OPERATION

D.C. and Operating Characteristics for Programming Operation

T_A = 25°C, V_{CC} = 0V, V_{BB} = +12V ± 10%, $\overline{CS}$ = 0V unless otherwise noted

SYMBOL	TEST	MIN.	TYP.	MAX.	UNIT	CONDITIONS
I_{LI1P}	Address and Data Input Load Current			10	mA	V_{IN} = −48V
I_{LI2P}	Program and V_{GG} Load Current			10	mA	V_{IN} = −48V
I_{BB}	V_{BB} Supply Load Current		.05		mA	
I_{DDP}(1)	Peak I_{DD} Supply Load Current		200		mA	V_{DD} = V_{prog} = −48V V_{GG} = −35V
V_{IHP}	Input High Voltage			0.3	V	
V_{IL1P}	Pulsed Data Input Low Voltage	−46		−48	V	
V_{IL2P}	Address Input Low Voltage	−40		−48	V	
V_{IL3P}	Pulsed Input Low V_{DD} and Program Voltage	−46		−48	V	
V_{IL4P}	Pulsed Input Low V_{GG} Voltage	−35		−40	V	

Note 1: I_{DDP} flows only during V_{DD}, V_{GG} on time. I_{DDP} should not be allowed to exceed 300mA for greater than 100μsec. Average power supply current I_{DDP} is typically 40mA at 20% duty cycle.

A.C. Characteristics for Programming Operation

$T_{AMBIENT}$ = 25°C, V_{CC} = 0V, V_{BB} = + 12V ± 10%, $\overline{CS}$ = 0V unless otherwise noted

SYMBOL	TEST	MIN.	TYP.	MAX.	UNIT	CONDITIONS
	Duty Cycle (V_{DD} , V_{GG})			20	%	
$t_{\phi PW}$	Program Pulse Width			3	ms	V_{GG} = −35V, V_{DD} = V_{prog} = −48V
t_{DW}	Data Set Up Time	25			μs	
t_{DH}	Data Hold Time	10			μs	
t_{VW}	V_{DD} , V_{GG} Set Up	100			μs	
t_{VD}	V_{DD} , V_{GG} Hold	10		100	μs	
t_{ACW} (2)	Address Complement Set Up	25			μs	
t_{ACH} (2)	Address Complement Hold	25			μs	
t_{ATW}	Address True Set Up	10			μs	
t_{ATH}	Address True Hold	10			μs	

Note 2. All 8 address bits must be in the complement state when pulsed V_{DD} and V_{GG} move to their negative levels. The addresses (0 through 255) must be programmed as shown in the timing diagram for a minimum of 32 times.

Switching Characteristics for Programming Operation

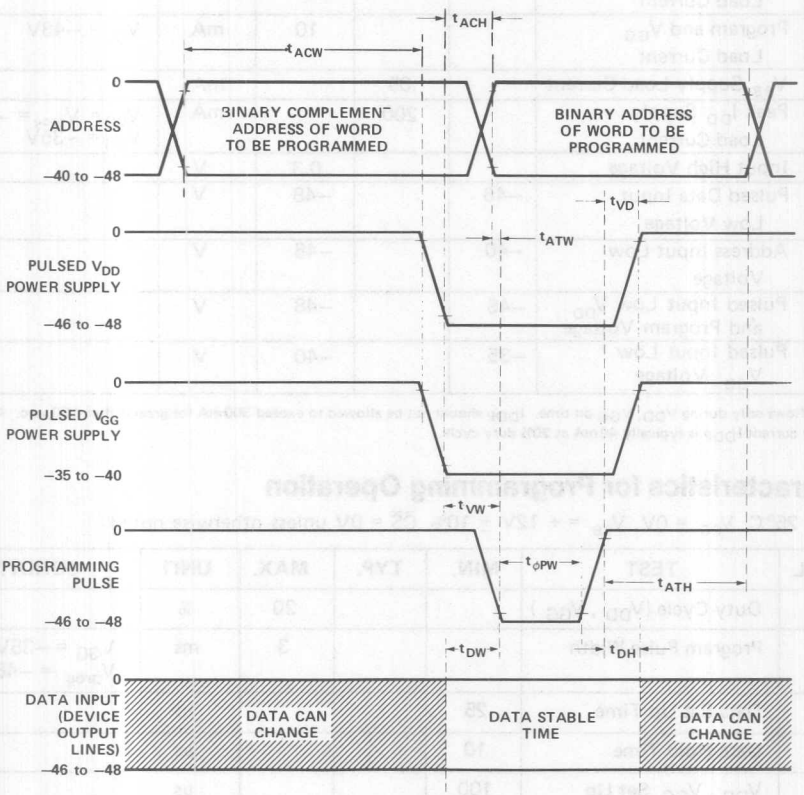
PROGRAM OPERATION

Conditions of Test:

Input pulse rise and fall times $\leq 1\mu$sec

$\overline{CS}$ = 0V

PROGRAM WAVEFORMS



Programming Operation

When the Data Input for the Program Mode is:	Then the Data Output during the Read Mode is:
V_{ILIP} = ~–48V pulsed	Logic 1 = V_{OH} = 'P' on tape
V_{IHP} = ~ 0V	Logic 0 = V_{OL} = 'N' on tape

WORD	ADDRESS							
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
|	|	|	|	|	|	|	|	|
|	|	|	|	|	|	|	|	|
255	1	1	1	1	1	1	1	1

Address Logic Level During Read Mode: Logic 0 = V_{IL} (~.3V) Logic 1 = V_{IH} (~3V)

Address Logic Level During Program Mode: Logic 0 = V_{IL2P} (~–40V) Logic 1 = V_{IHP} (~0V)

MCS-40™ Program Memory In 4702A PROMs

Memory address, memory data, I/O bus, and chip select lines from 4289 or a 4008/4009 are defined with respect to positive logic. The MCS-40™ data and control lines from the CPU are defined with respect to negative logic. As a result, in 4702A program memory used with the 4289 or 4009, programs should be coded with logic "1" = high level and logic "0" = low level (i.e., NOP = 0000 0000 = NNNN NNNN).

For 4702A PROM programs which are to be converted to 4001 or 4308 ROM memory, a preferred method is to use negative logic program memory in the 4702A and place inverting buffers at the data inputs of the 4289 or 4008/4009. This allows program code to be consistent with that of the 4001 and 4308 mask programmed ROMs and assures that 4289 or 4008/4009 input capacitance will not limit system speed when using several 4702A PROMs for program storage. (Note that programs are defined for the 4001/4308 ROMs in terms of negative logic such that NOP = 0000 0000 = PPPP PPPP.)

Programming Operation

I. Operation of the 4702A in Program Mode

Initially, all 2048 bits of the ROM are in the "0" state (output low). Information is introduced by selectively programming "1"s (output high) in the proper bit locations.

Word address selection is done by the same decoding circuitry used in the READ mode (see table on page 6 for logic levels). All 8 address bits must be in the binary complement state when pulsed V_{DD} and V_{GG} move to their negative levels. The addresses must be held in their binary complement state for a minimum of 25 μsec after V_{DD} and V_{GG} have moved to their negative levels. The addresses must then make the transition to their true state a minimum of 10 μsec before the program pulse is applied. The addresses should be programmed in the sequence 0 through 255 for a minimum of 32 times. The eight output terminals are used as data inputs to determine the information pattern in the eight bits of each word. A low data input level (−48V) will program a "1" and a high data input level (ground) will leave a "0" (see table on page 6). All eight bits of one word are programmed simultaneously by setting the desired bit information patterns on the data input terminals.

During the programming, V_{GG}, V_{DD} and the Program Pulse are pulsed signals.

II. Programming of the 4702A Using Intel Microcomputers

Intel provides low cost program development systems which may be used to program its electrically programmable ROMs. Note that the programming specifications that apply to the 4702A are identical to those for Intel's 1702A.

A. Intellec® 4

The Intellec® 4 program development system is used as a program development tool for the 4004 and 4040 microprocessors. As such, it is equipped with a PROM programmer card and may be used to program Intel's electrically programmable and ultraviolet erasable ROMs.

An ASR-33 teletype terminal is used as the input device. Through use of the Intellec software system monitor, programs to be loaded into PROM may be typed in directly or loaded through the paper tape reader. The system monitor allows the program to be reviewed or altered at will prior to actually programming the PROM. For more complete information on this program development system, refer to the Intel Microcomputer Catalog or the Intellec Specifications.

B. Intellec® MDS

An Intellec® MDS system can also be used with a Universal PROM Programmer (UPP) to program 4702A PROMs. The 1702A/4702A personality card must be plugged into the appropriate PROM programmer card socket of the UPP for this programming operation.

III. 4702A Erasing Procedure

The 4702A may be erased by exposure to high intensity short-wave ultraviolet light at a wavelength of 2537A. The recommended integrated dose (i.e., UV intensity x exposure time) is 6W-sec/cm^2. An example of an ultraviolet source which can erase the 4702A in 10 to 20 minutes is the Model S-52 short-wave ultraviolet lamp manufactured by Ultra-Violet Products, Inc. (5114 Walnut Grove Avenue, San Gabriel, California). The lamps should be used without short-wave filters, and the 4702A to be erased should be placed about one inch away from the lamp tubes.

MCS-40™ Program Memory in 4702A PROMs

Memory address, memory data, I/O bus, and chip select lines from 4289 or 4308 MCS-40™ devices are defined with respect to positive logic. The MCS-40™ data and control lines from the CPU are defined with respect to negative logic. As a result, in 4702A program memory used with the 4289 or 4308, programs should be coded with logic "1" = high level and logic "0" = low level (i.e., NOP = 0000 0000 = NNNN NNNN).

For 4702A PROM programs which are to be converted to 4001 or 4308 ROM memory, a preferred method is to use negative logic program memory in the 4702A and place inverting buffers at the data inputs of the 4289 or 4308. This allows program code to be consistent with that of the 4001 and 4308 mask programmed ROMs and assures that 4.3V on 4008/4009 input capacitance will not limit system speed when using several 4702A PROMs for program storage. (Note that programs for 4001/4308 ROMs in terms of negative logic such that NOP = 0000 0000 = FFFF FFFF.)

Programming Operation

I. Operation of the 4702A in Program Mode

Initially, all 2048 bits of the ROM are in the "0" state (output low). Information is introduced by selectively programming "1"s (output high) in the proper bit locations.

Word address selection is done by the same decoding circuitry used in the READ mode (see table on page 8 for logic levels). All 8 address bits must be in the binary complement state when pulsed V_{CC} and V_{DD} move to their negative levels. The addresses must be held in their binary complement state for a minimum of 23 μsec after V_{CC} and V_{DD} have moved to their negative levels.

The address must then make the transition to their true state a minimum of 10 μsec before the program pulse is applied. The address should be programmed in the sequence 0 through 255 for a minimum of 32 times. The eight output terminals are used as data inputs to determine the information pattern in the eight bits of each word. A low data input level (−48V) will program a "1" and a high data input level (ground) will leave a "0" (see table on page 8). All eight bits of one word are programmed simultaneously by setting the desired bit information patterns on the data input terminals.

During the programming, V_{CC}, V_{DD} and the Program Pulse are pulsed signals.

II. Programming of the 4702A Using Intel Microcomputers

Intel provides low cost program development systems which may be used to program its electrically programmable ROMs. Note that the programming specifications that apply to the 4702A are identical to those for Intel's 1702A.

A. Intellec® 4

The Intellec® 4 program development system is used as a program development tool for the 4004 and 4040 microprocessors. As such, it is equipped with a PROM programmer card and may be used to program Intel's electrically programmable and ultraviolet erasable ROMs.

An ASR-33 teletype terminal is used as the input device. Through use of the Intellec software system monitor, programs to be loaded into PROM may be typed in directly or loaded through the paper tape reader. The system monitor allows the program to be reviewed or altered at will prior to actually programming the PROM. For more complete information on this program development system, refer to the Intel Microcomputer Catalog or the Intellec Specifications.

B. Intellec® MDS

An Intellec® MDS system can also be used with a Universal PROM Programmer (UPP) to program 4702A PROMs. The 1702A/4702A personality card must be plugged into the appropriate UPP for the programming operation.

III. 4702A Erasing Procedure

The 4702A may be erased by exposure to high intensity short-wave ultraviolet light at a wavelength of 2537Å. The recommended integrated dose (i.e., UV intensity x exposure time) is 6W-sec/cm^2. An example of an ultraviolet source which can erase the 4702A in 20 minutes is the Model S-52 short-wave ultraviolet lamp manufactured by Ultra-Violet Products, Inc. (San Gabriel, California). The lamp should be used without short-wave filters, and the 4702A to be erased should be placed about one inch away from the lamp tubes.

CHAPTER 6.
MCS-40™ APPLICATION NOTES

This chapter contains a series of application notes on various MCS-40 devices. The "4265 Application Examples" note gives a brief description of the 4265 Programmable General Purpose I/O device and then follows with a number of typical systems using 4265s. The 4265 GP I/O's capability to interface to standard RAM memories such as the 5101 CMOS RAM and the 2111 MOS RAM is illustrated; interfaces to 8-bit I/O peripheral devices such as the 8251 Programmable Serial Communications device and the 8253 Programmable Interval Timer as well as 8-bit CPUs are illustrated; standard general purpose I/O interfaces are described.

The "4269 Applications Examples" note gives a brief description of the 4269 Programmable Keyboard/Display device. Typical systems are illustrated for the various keyboard and display modes of the 4269. A detailed logic diagram including all necessary external part number information is provided to illustrate a complete keyboard/display system implemented with a 4269.

The note "Non-Volatile Memory Using the MCS-40 with the 5101 RAM" explains how a CMOS 5101 memory can be interfaced via the 4289 Standard Memory Interface device to implement a non-volatile memory system. A battery subsystem is used to maintain information should a power failure occur. The software used in implementing this system is described in detail.

4265 APPLICATION EXAMPLES

- □ Multi-Mode 14 Operating Modes
- □ System Compatible With 4289, 4308, and 4001
- □ 16 Lines of I/O Capability
- □ Bit Set/Reset
- □ Multiplexable Outputs
- □ Eight Bit Transfer Mode
- □ Interfaces to 8080 Peripherals
- □ Synchronous and Asynchronous Interface
- □ Strobed Buffer Inputs and Outputs
- □ TTL Interface
- □ Up to Eight 4265s Per System
- □ Interface to Standard RAMs
- □ Directly MCS-40™ Bus Compatible

Introduction

The 4265 is a general purpose I/O device designed to interface with the MCS-40™ microcomputer family. This device provides four software programmable 4-bit I/O ports which can be configured to allow any of fourteen unique operating modes for interfacing to data memory or a variety of user peripheral devices.

A single MCS-40 system can accommodate up to four 4265s (one per CM-RAM) without external logic or up to eight 4265s with one external decoder.

The 4265 resides on the MCS-40 Data Bus and uses the same selection procedure as a 4002 RAM device. The upper two bits of the SRC address which appear on data bus lines D3 and D2 during X2 of the SRC instruction are compared with an internal 2-bit code. A valid compare selects the 4265 for MCS-40 I/O commands. As in the case of the 4002 or any MCS-40 peripheral circuit, selection occurs only when the proper SRC code and the CM signal are present simultaneously.

The 4265 provides an extremely flexible, general purpose I/O system capable of handling 4 or 8-bit input or output data. One of fourteen basic operating modes can be selected (software programmable) as described below.

The MCS-40 microprocessor system contains a group of 16 I/O instructions. Of these 16 instructions, two (RDR, WRR) are used for ROM port communication and one (WPM) is used for program memory operations. This leaves 13 instructions available for control and data transfer functions.

Control Functions: Two types of operations are possible with the 4265. First, the device (once selected) can be programmed to one of fourteen basic operating modes. This is accomplished by executing a WMP instruction which sends the 4-bit contents of the CPU's Accumulator to the 4265 where it is decoded and used to logically configure the device. A second control operation makes use of the WRM instruction to select one of eight output lines (Port Y or Z) and to perform a SET or RESET operation on that line. This is accomplished by interpreting the 4-bit Accumulator value as follows: The upper three bits select one of eight output latches; the least significant bit determines whether a SET or RESET operation is to be performed.

Data Transfer Functions: The remaining eleven instructions provide four WRITE operations (WR0, WR1, WR2, WR3) and seven READ operations (RD0, RD1, RD2, RD3, ADM, SBM, RDM). These allow data in 4-bit or 8-bit format to be transmitted between the 4265 and external I/O devices or memory devices (all transfers between processor and 4265 are 4-bit transfers).

Hardware Description

The 4265 is packaged in a 28-pin DIP. The pin configuration is shown and a functional description of each pin is given below:

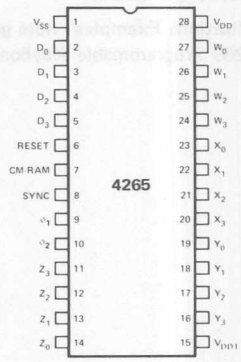


Figure 1. Pin Configuration.

PIN DESCRIPTION

Pin No.	Designation	Function
2–5	D0–D3	Bidirectional data bus. All address, instruction and data communication between the CPU and I/O ports are transferred on this port.
6	RESET	A negative "1" level (V_{DD}) applied to this pin clears all storage elements, places the 4265 in the Reset Mode, and deselects the device.
7	CM	Command input driven by a CM-RAM output of the processor. Used for decoding SRC, RDM, WRM, WMP, SBM, ADM, WR0–3 and RD0–3.
8	SYNC	Synchronization signal generated by the processor; indicates the beginning of an instruction.
9,10	ϕ_1–ϕ_2	Non-overlapping clock signals which determine timing.
24–27 20–23 16–19 11–14	W3–W0 X3–X0 Y3–Y0 Z3–Z0	Four programmable I/O ports having different functional designation depending on 4265 mode of operation. A data bus "1" negative true (V_{DD}) will appear on a port as a "1" positive true (V_{SS}). These ports are TTL compatible. (Ports W, X, and Y low-power TTL compatible.)
28	V_{DD}	Main power supply pin. Value must be V_{SS} –15V ±5%.
15	V_{DD1}	Supply voltage for I/O ports.
1	V_{SS}	Most positive supply voltage (V_{DD1} = 0V, V_{SS} = 5V for TTL I/O ports).
28 = TOTAL PINS		

MODE Description

Shown below in Table I is a summary of each of the operating and control modes of the 4265.

Table I. 4265 Programmable Modes

OPERATING MODES

- Mode 1 – 8-Bit Asynchronous I/O Port (Bidirectional)
  4-Bit Input Port (Unbuffered)
- Mode 2 – 8-Bit Asynchronous I/O Port (Bidirectional)
  4-Bit Output Port
- Mode 3 – 8-Bit Synchronous I/O Port (Bidirectional)
  4-Bit Synchronous Output Port
- Mode 4 – Four 4-Bit Output Ports
- Mode 5 – Three 4-Bit Output Ports
  One 4-Bit Input Port (Unbuffered)
- Mode 6 – Two 4-Bit Output Ports
  Two 4-Bit Input Ports (Unbuffered)
- Mode 7 – One 4-Bit Output Port
  Three 4-Bit Input Ports (Unbuffered)
- Mode 8 – Three 4-Bit Synchronous Output Ports
- Mode 9 – Two 4-Bit Synchronous Output Ports
  One 4-Bit Asynchronous Input Port

OPERATING MODES

- Mode 10 – One 4-Bit Synchronous Output Port
  Two 4-Bit Asynchronous Input Ports
- Mode 11 – Three 4-Bit Asynchronous Input Ports
- Mode 12 – 8-Bit Address Port
  4-Bit Synchronous I/O Port (Bidirectional)
  2 Device Selection Control Signals
- Mode 13 – 8-Bit Address Port
  4-Bit Asynchronous I/O Port (Bidirectional)

CONTROL AND OPERATING MODE

- Mode 0 – Four 4-Bit Input Ports (Unbuffered)
  Resets I/O Buffers

CONTROL MODES

- Mode 14 – Disables all output buffers, allowing another 4265 to be multiplexed at the port level.
- Mode 15 – Enables output buffers, previous mode restored.

4265 I/O Instructions

Shown below in Table II is a summary of MCS-40 I/O instructions which are used with the 4265 Programmable General Purpose I/O Device.

Table II. 4265 Input/Output Instructions

Hex Code	MNEMONIC	OPR D_3 D_2 D_1 D_0	OPA D_3 D_2 D_1 D_0	DESCRIPTION OF OPERATION		
Mode Independent Operations						
E0	WRM	1 1 1 0	0 0 0 0	The port Y or port Z bit designated by D_3 D_2 D_1 of the accumulator is set or reset according to D_0 (1=set, 0=reset).[1]		
E1	WMP	1 1 1 0	0 0 0 1	Sets the mode of the 4265 to the value contained in the accumulator.[2]		
Mode Dependent Operations						
				Mode 1-3	**Mode 0, 4-11**	**Mode 12 and 13**
2-	SRC	0 0 1 0	R R R 1	For modes 0-11, the contents of register pair RRR are used to select the 4265 chip (first two bits of first register will contain 10 or 11, depending on chip address)		(RRR_{even})→ Port W (RRR_{odd})→ Port X
E4	WR0	1 1 1 0	0 1 0 0	(ACC)→ Port W	(ACC)→ Port W[1]	(ACC)→ Port Y
E5	WR1	1 1 1 0	0 1 0 1	(ACC)→ Port X	(ACC)→ Port X[1]	(ACC)→ Port Y
E6	WR2	1 1 1 0	0 1 1 0	(ACC)→ Port Y[1]	(ACC)→ Port Y[1]	(ACC)→ Port Y
E7	WR3	1 1 1 0	0 1 1 1	—	(ACC)→ Port Z[1,3]	(ACC)→ Port Y
EC	RD0	1 1 1 0	1 1 0 0	(Port W)→ ACC	(Port W)→ ACC	(Port Y)→ ACC
ED	RD1	1 1 1 0	1 1 0 1	(Port X)→ ACC	(Port X)→ ACC	(Port Y)→ ACC
EE	RD2	1 1 1 0	1 1 1 0	(Port Y)→ ACC	(Port Y)→ ACC	(Port Y)→ ACC
EF	RD3	1 1 1 0	1 1 1 1	(Port Z)→ ACC	(Port Z)→ ACC	(Port Y)→ ACC
E9	RDM	1 1 1 0	1 0 0 1	(Port Y)→ ACC	(Port Y)→ ACC	(Port Y)→ ACC
EB	ADM	1 1 1 0	1 0 1 1	(Port Y)+(ACC) +CY→ACC	(Port Y)+ACC +CY→ACC	(Port Y)+ACC +CY→ACC
E8	SBM	1 1 1 0	1 0 0 0	(ACC)−(Port Y) −CY→ACC	(ACC)−(Port Y) −CY→ACC	(ACC)−(Port Y) −CY→ACC

NOTES:
1. Action if Port is designated as Output Port; otherwise, no action.
2. WMP 1110 disables all I/O ports. WMP 1111 enables all I/O ports. In both cases, the mode is not changed.
3. No action in Modes 8-11.

APPLICATION EXAMPLES

In the following section, various MCS-40 systems utilizing 4265 General Purpose I/O Devices are described. While these applications are only a sample of a wide variety of possible system configurations, they should serve to illustrate the main features of the 4265, e.g., the 8-bit I/O device interface capability, the external RAM memory interface capability, and the bit set/reset capabilities.

Interfacing the 8251 USART to the MCS-40™ via the 4265

MCS-40 applications very often require an asynchronous serial communication link. The communications can be local over a short distance using a twist pair wire link, or over long distance using a MODEM. The 8251 provides the parallel-to-serial conversion and the communication discipline required to implement a communication scheme. The 8251 requires data and control from the MCS-40 and provides status. This information is routed to the MCS-40 via the 4265, Mode 3, Port W, X. Port Z provides the control for the information transfer (see Figure 2). When the 8251 transmitter is either empty (TxE) or Ready (TxR), or the Receiver is Ready (RxR), the appropriate 8251 lines will become active. These lines can be ORed to Interrupt the 4040 CPU or can be polled via an Input Port. A status read of the 8251 will also input the above signals for interrogation.

8251 USART (Asynchronous Mode)

Data	Control	Status
	Baud Rate	Framing Error
Read USART	1x	Overrun Error
Write USART	16x	Parity Error
	64x	Tx Empty
	Character Length	Rx RDY
	5-bit	Tx RDY
	6-bit	DSR
	7-bit	
	8-bit	
	Parity Enable	
	Parity Even/Odd	
	Stop Bit	
	1 Stop Bit	
	1½ Stop Bit	
	2 Stop Bit	
	Reset	
	Modem Control	

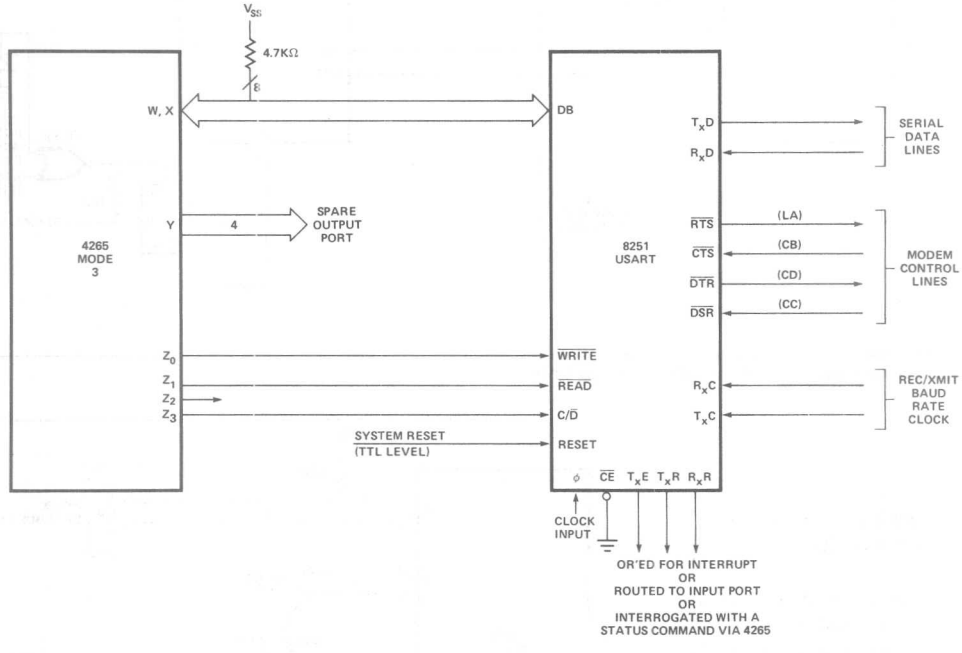


Figure 2. Interfacing the 8251 USART to the MCS-40™ via the 4265.

4265 Interfaced to 3214 Priority Interrupt Unit

An MCS-40 system can be utilized in a multi-interrupt environment. If several devices external to the MCS-40 can request interrupt simultaneously, it becomes necessary to establish a priority scheme for arbitrating and servicing the interrupting devices. The 3214, to which a 4265 can interface, has the ability to prioritize the interrupts and assign an address to each interrupt input. This address can be used to indirectly vector the CPU to the program of the interrupt service routine for the interrupting device. Upon detecting a valid interrupt request, the 3214 will provide a 3-bit address on Port W of the 4265 and place the 4040 INT (interrupt) line low after the D flip-flop is clocked by the IA output of

the 3214 (see Figure 3). The 4040 will generate an INT ACK which will clear the D flip-flop. The IA output of the 3214 will be cleared by the ϕ_{IT} clock. The 4040 is then ready to interrogate the 3214 Address on Port W and jump to the interrupt routine. Prior to returning from the interrupt routine, the ENLG line may be interrogated to see if any other interrupts are pending. This procedure is optional. A reloading of the 3214 status level via Port X and Z_0 will enable the 3214 for additional interrupt requests.

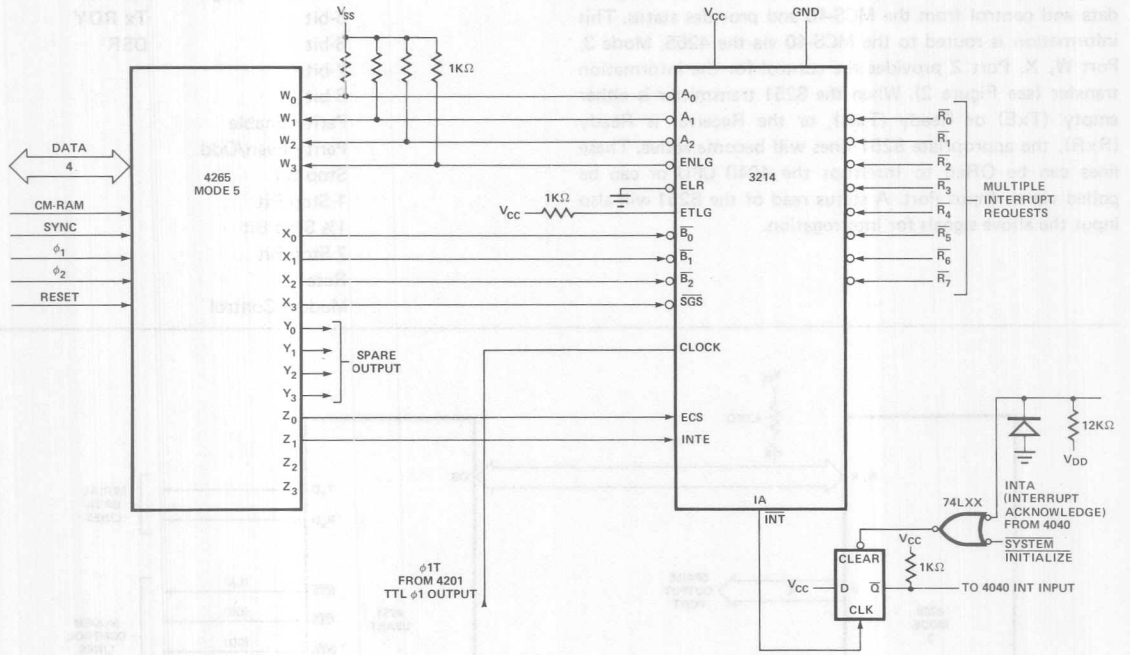


Figure 3. 4265 Interfaced to 3214 Priority Interrupt Unit.

Interfacing the MCS-80™* to the MCS-40™ via the 4265

The 4265 can reside directly on the MCS-80 data and control bus (see Figure 4). As such it provides a direct interface between the MCS-40 and MCS-80 microcomputer systems. The ability to interface one or more MCS-40 systems to an MCS-80 system allows the designer to construct very powerful distributed intelligence microcomputer systems with a minimum of complexity.

*MCS-80™ is an Intel trademark.

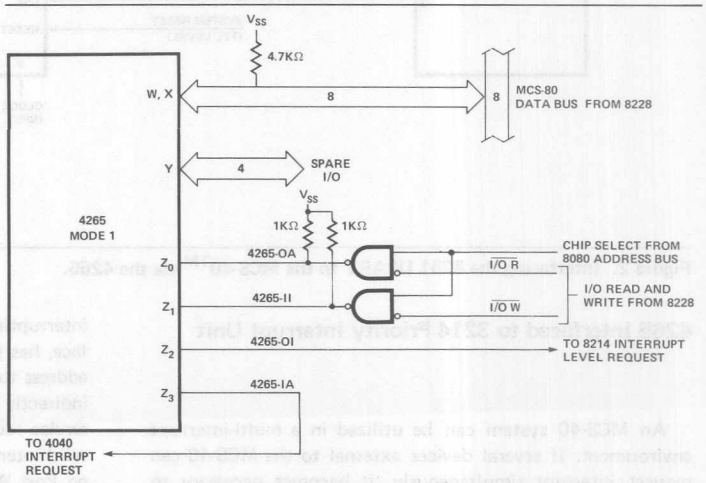


Figure 4. Interfacing the MCS-80 to the MCS-40 via the 4265.

Interfacing the 8253 Programmable Interval Timer to the MCS-40™ System via the 4265

The 8253 is a multi-mode, programmable interval timer functionally equivalent to an array of one shots, programmable counters, and/or rate selectors. Each 8253 contains three independent counters. Figure 5 illustrates how a 4265 interfaces to an 8253.

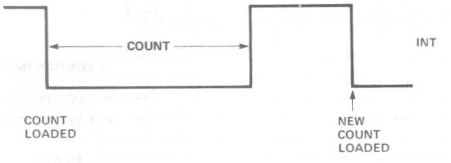
Counter Operation

a. Each counter can count in binary or BCD.

b. Each counter will have the following pins: clock/event count, output/interrupt and gate/trigger/reset.

c. Each counter can operate in the following modes:

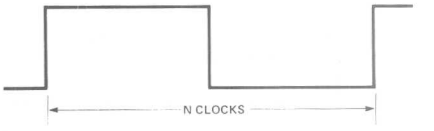
1. Interrupt on terminal count – When count is loaded, the output (interrupt) will go from high to low for the duration of the count. The gate will enable counting. This mode is used for counting or sampling events.



2. Retriggerable one-shot – Active low output for duration of count. Initiated by rising edge of gate.

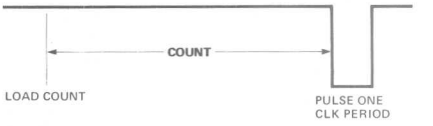
3. Rate Generator – Divide-by-N counter, reset with low on gate. The output will be low for one input clock period for each N clocks.

4. Rate generator with square wave output – Same as 3 above, except a 50% duty cycle square wave is generated for each N clocks.



5. Dual one-shot (non-retriggerable) – Output will be high. When count is loaded, counter will start counting. When terminal count is reached, the output will go low for one clock period. The gate inhibits count.

The count and control modes are loaded via the 4265. Port Z_2 and Z_3 lines steer the information to the desired counter.



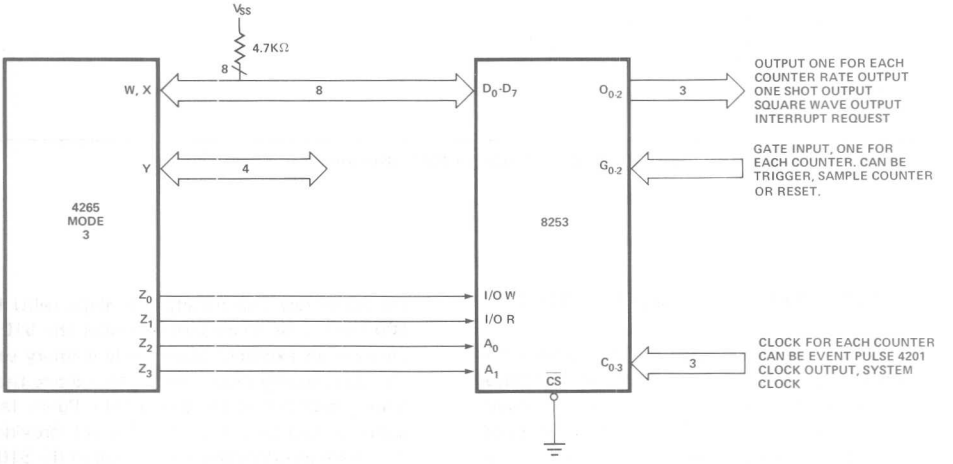


Figure 5. Interfacing the 8253 Programmable Interval Timer to the MCS-40™ System via the 4265.

Interfacing the 4265 to a Multiple of 1K x Banks of RAM Memory

The 4265 in MODE 12 can directly interface to 1K X 4 bits of RAM memory. This amount of data storage is generally adequate for most MCS-40 systems. However, some applications do require larger arrays of memory. The scheme shown in Figure 6 illustrates how the 4265 MODE 12 operation can be expanded to access sixteen 1K X 4-bit banks of RAM (using 2111 256 x 4 RAM chips in this example).

The Port W, X and Z_2, Z_3 lines provide the 10-bit address for a 1K selection. The Z_2, Z_3 lines are decoded using one-half of a dual 1 of 4 decoder DIP or equivalent. This selects one of the four 2111 RAM elements. Port W, X is the common 8-bit address bus used to select the 1 of 256 RAM location. Read (input) or write (output) data is transferred on a common bus between the RAM element and the 4265. The 4265 provides the 2111 with the basic control signals for read and write (R/W and OD) operations.

This basic scheme can be expanded to allow the single 4265 to address multiple 1K X 4 pages of RAM. The 2111 has two chip select lines. One chip select line is dedicated to a bank select. If sixteen banks were desired, an additional 4-bit output port and a 1 of 16 decoder would provide the bank selection hardware. The additional port can be obtained from another 4002, 4265, or a 4308.

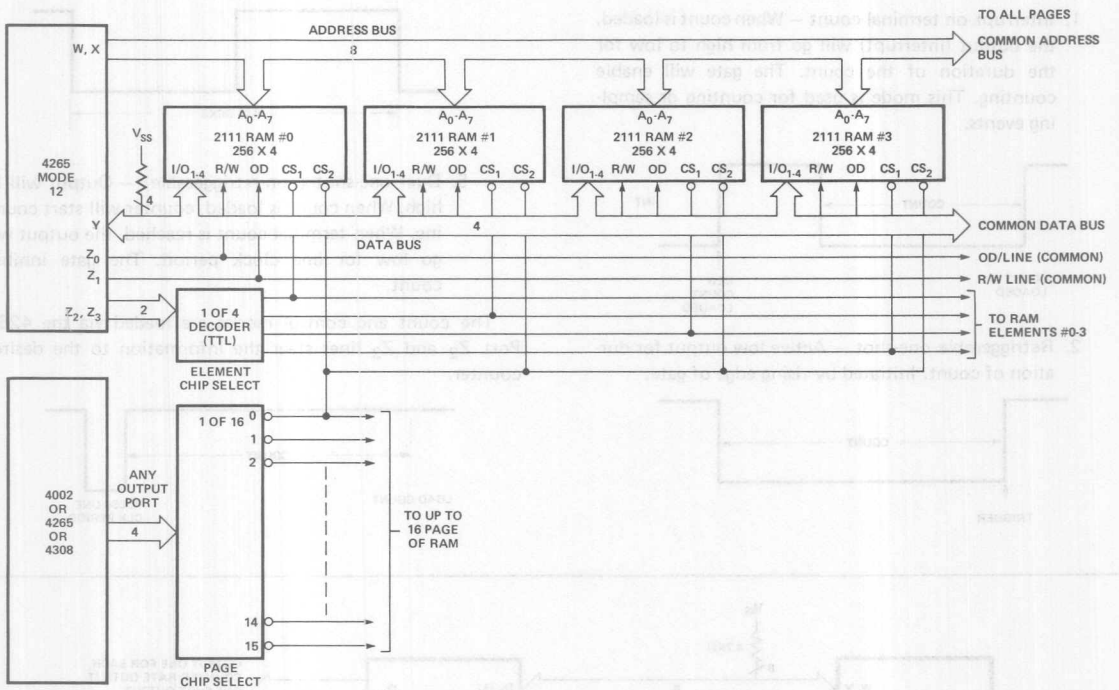


Figure 6. Interfacing the 4265 to a Multiple of 1K X 4 Banks of RAM Memory.

Using the 5101 RAM as Data Storage for MCS-40™

The 5101 CMOS RAM can be used as data storage for an MCS-40 system in the same way as the 2111/2101 is utilized (see Figure 7). However, because of the ultra low standby power requirement of the 5101, it becomes very advantageous to use this RAM when information must be retained after a power drop. The data in a 5101 can be preserved with the use of low cost batteries (penlight cells) for greater than 400 days. This characteristic makes the 5101 battery combination an excellent non-volatile memory substitute. When the main supply drops, the battery diode becomes forward-biased, maintaining the 5101s only. Power fail detection circuitry should be provided which will provide the Low V_{CC} Data Retention Waveform described in the 5101 specification in Chapter 5.

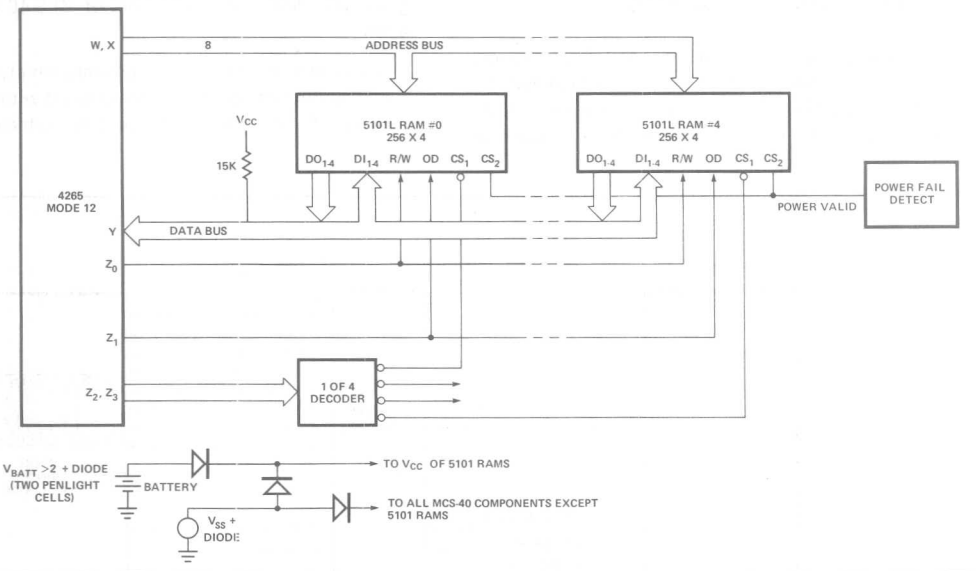


Figure 7. Using the 5101 RAM as Data Storage for MCS-40.

4265 Interface to Daisy Wheel Printer

The 4265 can be used to interface to an array of printer mechanisms. The daisy wheel printer is one such mechanism (see Figure 8). The 4265 mode 8 is used to provide vertical, horizontal and character data to the printer. Port Z provides the strobe command for the printer. To obtain mutually exclusive strobes (e.g., one strobe per operation over Ports W, X and Y), the 4265 is first disabled with a WMP mode 14. All but the last port (the port associated with the strobe) is loaded with data. The 4265 is then enabled with a WMP mode 15 and the last port is written, generating the strobe. If just a paper feed of lines is desired, the 4265 would be disabled, Ports W and X would be loaded via a WR0, WR1 command sequence, respectively, the 4265 would be enabled and Port Y would be loaded via WR2. This would cause the PAPER FEED STROBE to be active. The 4265 can be disabled and enabled alternately without changing the basic mode of operation. The 4265 output buffers can be loaded in both states.

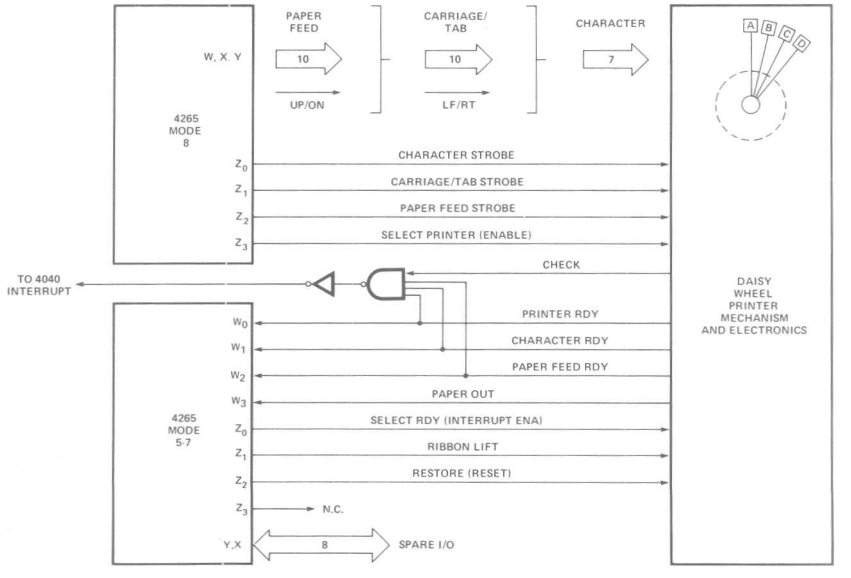


Figure 8. 4265 Interface to Daisy Wheel Printer.

Drum Printer Interface with the 4265

Popular drum printers can be interfaced to MCS-40 using one 4265. Figure 9 shows an interface to a 12 column, 16 ROW drum printer. The drum printer timing is well within the MCS-40 scope of operation. A typical drum printer program takes less than ninety lines of MCS-40 code using the 4265.

Expanding the number of columns simply requires additional output lines which can be obtained with the 4265. The number of rows can be increased by software techniques.

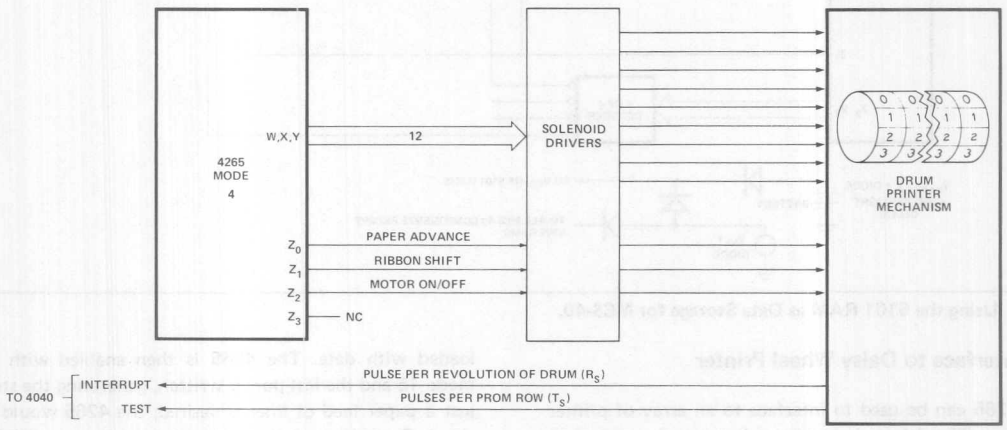


Figure 9. Drum Printer Interface with the 4265.

4269 APPLICATION EXAMPLES

4269 KEYBOARD FEATURES:

- **Programmable to Interface to Encoded Keyboard (8-Bit Code), 64-Key Scanned Keyboard (Expandable to 128 Keys) or Sensor Matrix (64 Sensors)**
- **8 Character FIFO Character Buffer (or RAM in Sensor Mode)**
- **2 Key Rollover and Key Debounce**
- **External Interrupt Line to Indicate When a Character Has Been Entered in Character Buffer**

4269 DISPLAY FEATURES:

- **Programmable to Interface to Individually Scanned Displays or Burrough's Self-Scan* Drive (16, 18, or 20 Characters)**
- **Two 16 x 4 Display Registers Recirculated Synchronously With Keyboard Scan Lines to Give Automatic Display Refresh**
- **Display Registers Loadable and Readable Selectively or Sequentially**
- **40 Pin Dual In-Line Package**

Introduction

The 4269 Programmable Keyboard/Display (PKD) device provides an intelligent interface between an MCS-40 CPU and the keyboard and display portions of an MCS-40 design. The 4269's functions thus allow the use of sophisticated keyboards and displays without placing a large load on the CPU.

The MCS-40 data bus will provide the path for information transfer between the PKD and the 4040 or 4004 CPU. The PKD can be programmed to operate in one of three input modes and one of four output modes as defined by an instruction from the CPU. The modes are:

Input
- Sensor, Scanned
- Keyboard, Scanned
- Encoded Keyboard

Output
- Individually Scanned Display Drive
- Self-Scan Drive: 16 Characters
  18 Characters
  20 Characters

The 4269 resides on a CM-RAM line of an MCS-40 system and has a fixed RAM address, #1. Hence, there can be up to four PKD per system without additional logic, one per CM-RAM. The PKD can be accessed with the MCS-40 I/O instruction set to interrogate the keyboard buffer FIFO/sensor RAM and load or read the display registers.

Hardware Description

The 4269 is packaged in a 40 pin DIP. The pin configuration is shown below and a functional description of each pin is given.

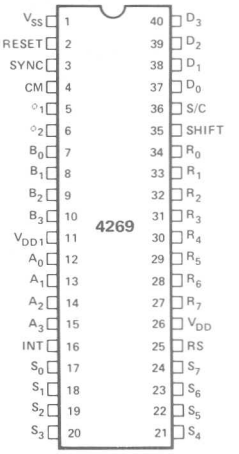


Figure 1. Pin Configuration

*Self-Scan is a registered trademark of the Burroughs Corporation.

PIN DESCRIPTION

Pin No.	Designation	Function
37-40	D0-D3	Bi-directional data bus. All address, instruction and data communication between the CPU and the PKD are transmitted on these 4 pins.
5-6	ϕ_1-ϕ_2	Non-overlapping clock signals which are used to generate the basic chip timing.
2	RESET	RESET input. A low level (V_{DD}) applied to this input resets the PKD.
1	V_{SS}	Most positive supply voltage.
26	V_{DD}	Main power supply pin. Value must be V_{SS} - 15V ±5%.
3	SYNC	Synchronization input signal driven by SYNC output of the CPU.
4	CM	Command input driven by a CM-RAM output of processor.
17-24	S0-S7	These pins are scan outputs which are used for driving either the key switch or sensor matrix and/or for strobing the display digits. Each line is mutually exclusive, active high (V_{SS}), open drain.
25	RS	The RS pin is toggled for each complete scan of the S drive. This allows for the scan of 16 digits of display data. RS=V_{SS} for the last 8 digits. This line is open drain.
12-15 7-10	A0-A3 B0-B3	These two ports provide two 16 x 4 recirculating display register outputs which are synchronized to the S drive scan. In the gas discharge display mode, A3 is reset and A2 is the clock to the gas discharge display. The 16, 18, or 20 recirculating data characters (6 bits wide) are not synchronized with the S drive scan in the gas discharge mode. These lines are active high.
34-28	R0-R7	These pins are the return sense inputs which are connected to the 8 drive lines via the scanned key or sensor matrix. They are pulled to a low state (V_{DD}) in the sensor mode, pulsed low (V_{DD}) in the scanned keyboard mode, and pulled high upon switch closure. They are floating in the encoded keyboard mode.
35	SHIFT	This is the shift input. It is active high (V_{SS}). This pin is functional only in the scanned keyboard mode.
16	INT	This output is used to indicate when a keyboard or sensor character has been entered into the buffer. It is active low (V_{DD1}), open-sourced and may be "OR" -ed with other 4040 interrupt inputs.
11	V_{DD1}	Supply voltage for display register ports A and B and INT.
36	S/C	This pin is the control key input from the keyboard in the scanned mode. In encoded keyboard mode, this pin is used to input the strobe pulse from an external keyboard encoder. The strobe should be an active high pulse.

4269 Input/Output Instructions

Table I shows a summary of the MCS-40 I/O instructions which are used with the 4269 Programmable Keyboard/Display device.

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
E4	WRO	1 1 1 0	0 1 0 0	Set the input mode and output mode of the 4269 according to the value contained in the accumulator as follows: $D_3 D_2$ — $D_1 D_0$ 00 Individual, Scanned Displays, 8 or 16 Characters — 00 Sensor, Scanned 01 Gas Discharge, 20 Characters — 01 Contact Keyboard, Scanned 10 Gas Discharge, 18 Characters — 10 Encoded Keyboard, not Scanned 11 Gas Discharge, 16 Characters — 11 Not Used
2 -	SRC	0 0 1 0	R R R 1	The 1st two bits of the contents of the register pair RRR are used to select the 4269 (they will be 01, the binary address of a 4269). The remaining bits will be interpreted as follows, depending on the mode set: $\frac{RRR_{even}}{D_3D_2D_1D_0}$ $\frac{RRR_{odd}}{D_3D_2D_1D_0}$ **For gas discharge mode:** 0100 $n_3n_2n_1n_0$ — Selects the nth display register character of Display Register A with display outputs continuing to output the contents of Display Registers A and B. 0101 $n_3n_2n_1n_0$ — Selects the nth display register character of Display Register B with the display outputs continuing to output the contents of Display Registers A and B. 0110 $n_3n_2n_1n_0$ — Selects the nth display register character of Display Register A and blanks the A and B display output (with hex 20). 0111 $n_3n_2n_1n_0$ — Selects the nth display register character of Display Register B and blanks the A and B display output (with hex 20). **For individual, scanned display mode:** 0100 $n_3n_2n_1n_0$ — Selects one of 16 display register characters of Display Register A with the A output lines outputting display characters synchronized with the S Scan lines. 0101 $n_3n_2n_1n_0$ — Selects one of 16 display register characters of Display Register B with the B output lines outputting display characters synchronized with the S Scan lines. 0110 $n_3n_2n_1n_0$ — Selects one of 16 display register characters of Register A with Register A output lines being placed at Vss level. 0111 $n_3n_2n_1n_0$ — Selects one of 16 display register characters of Register B with Register B output lines being placed at Vss level. **For scanned sensor mode:** 0100 $n_3n_2n_1x$ — n_3-n_1 indicates an 8-bit sensor group to be read **For scanned keyboard or non-scanned encoded keyboard:** 01XX XXXX — SRC used only to select 4269
E7	WR3	1 1 1 0	0 1 1 1	Clears the keyboard/sensor and display logic and fills the display RAM with all blanks. (Does not change mode).

Table I. Part 1 4269 Input/Output Instructions

Hex Code	MNEMONIC	OPR D_3 D_2 D_1 D_0	OPA D_3 D_2 D_1 D_0	DESCRIPTION OF OPERATION
For gas discharge display mode:				
E5	WR1	1 1 1 0	0 1 0 1	Resets the internal display register pointer to Display Register A position 0 and forces the Display Registers to the blank code.
For individual, scanned display mode:				
E5	WR1	1 1 1 0	0 1 0 1	Resets the internal display register pointer to 0 and forces Display Registers to the blank code. Upper two bits of ACC select length of display as follows: D_3 0 Display B is 16 nibbles deep 1 Display B is 8 nibbles deep D_2 0 Display A is 16 nibbles deep 1 Display A is 8 nibbles deep
For individual, scanned display mode and gas discharge display mode:				
E0	WRM	1 1 1 0	0 0 0 0	Loads the contents of the register addressed by the internal display register pointer with the contents of ACC; then advances the displayed data by one digit in relation to the scan line timing, and increments the display register pointer. In the gas discharge mode, the display register pointer alternates between the A and B registers.[1]
E9	RDM	1 1 1 0	1 0 0 1	Loads ACC with the contents of the register addressed by the display register pointer and then increments the display register pointer. In the gas discharge mode, the display register pointer alternates between the A and B registers.[1]
E1	WMP	1 1 1 0	0 0 0 1	Loads the contents of the register addressed by the display register pointer with the contents of ACC.
EF	RD3	1 1 1 0	1 1 1 1	Loads ACC with the contents of the display register pointed to by the display register pointer.
EB	ADM	1 1 1 0	1 0 1 1	Adds the contents of the display register pointed to by the display register pointer to the accumulator with carry.
E8	SBM	1 1 1 0	1 0 0 0	Subtracts the contents of the display register pointed to by the display register pointer from the accumulator with borrow.
For keyboard input modes:				
E6	WR2	1 1 1 0	0 1 1 0	Clears FIFO/RAM logic, the status buffer, and the INT line.
ED	RD1	1 1 1 0	1 1 0 1	Reads the first nibble of the current FIFO register position.
EE	RD2	1 1 1 0	1 1 1 0	Reads the second nibble of the current FIFO register position. FIFO register position is then incremented to the next position.
EC	RD0	1 1 1 0	1 1 0 0	Loads ACC with the FIFO status.
For scanned sensor mode:				
E6	WR2	1 1 1 0	0 1 1 0	Clears the FIFO/RAM logic and the INT line.
ED	RD1	1 1 1 0	1 1 0 1	Loads into ACC the upper 4 bits of the 8-bit sensor RAM group previously addressed by an SRC instruction.
EE	RD2	1 1 1 0	1 1 1 0	Loads into ACC the lower 4 bits of the 8-bit sensor RAM group previously addressed by an SRC instruction.

Note 1: Alternation pattern is A_0, B_0, A_1, B_1, A_2, etc.

Table I. Part 2 4269 Input/Output Instructions

APPLICATION EXAMPLES

In the following section, various MCS-40 systems utilizing the 4269 Programmable Keyboard/Display device are described. These systems illustrate several of the many possible display and keyboard interfaces. Note the system shown in Figure 10 in particular. This diagram illustrates a 4269 interfaced to a complete keyboard and display system. The associated display drivers and required logic are also detailed.

4269 Keyboard Interface to Scanned Matrix Keyboard

The keyboard portion of the 4269 PKD can be set to the Scanned Keyboard mode to interface to a variety of contact or matrix keyboards. The Scanned Keyboard mode can directly scan up to 64 keys as is shown in Figure 2, which illustrates a 4269 keyboard interface to a 64-key momentary key switch type keyboard. Two-key rollover and 11ms key debounce are provided by the 4269 in this mode.

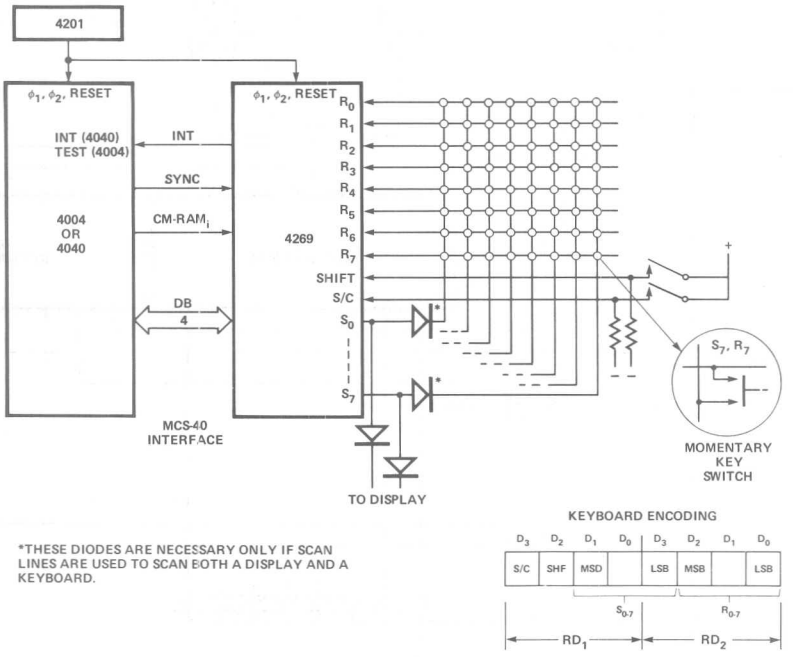


Figure 2. Matrix Keyboard.

4269 Keyboard Interface to a 128-Key Input

Figure 3 illustrates a 4269 keyboard interface expanded to 128 key scanned input. Note the diode arrangement which results a value of "0" being inputted in the shift input for keys from key matrix #1 and a "1" being inputted from key matrix #0. Debounce and 2-key rollover functions are provided for each 64-key matrix in such a configuration. However, these functions are not provided between the two matrices.

As an alternative to the implementation shown in Figure 3, two 4269 PKDs can be used for interfacing to 128 keys.

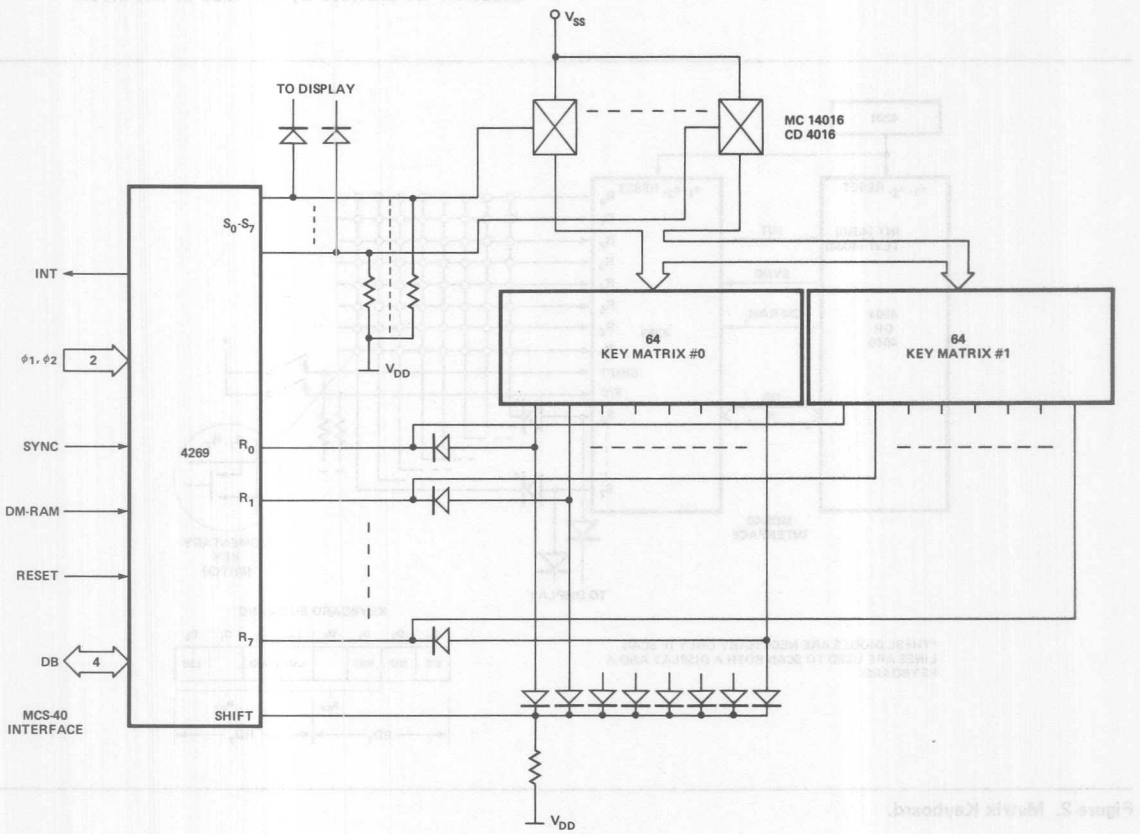


Figure 3. 128 Scanned Input Keys.

4269 Keyboard Interface to Encoded Keyboard

The keyboard portion of the 4269 PKD can also be set to the Encoded Keyboard mode to interface to a variety of encoded keyboards. The encoded keyboards strobe in their encoded key value via the S/C input of the 4269. Examples of such encoded keyboards are the Honeywell Micro Switch® and Lycon Keyboards. Figure 4 illustrates a 4269 Micro Switch® interface. Note that an encoded keyboard may have N-key or 2-key rollover as an internal feature of the keyboard itself.

Note also that, in addition to an interface to encoded keyboard, the Encoded Keyboard mode may be used as a general purpose strobed 8-bit input port. The internal keyboard buffer can store up to 8 sets of 8-bit input data values.

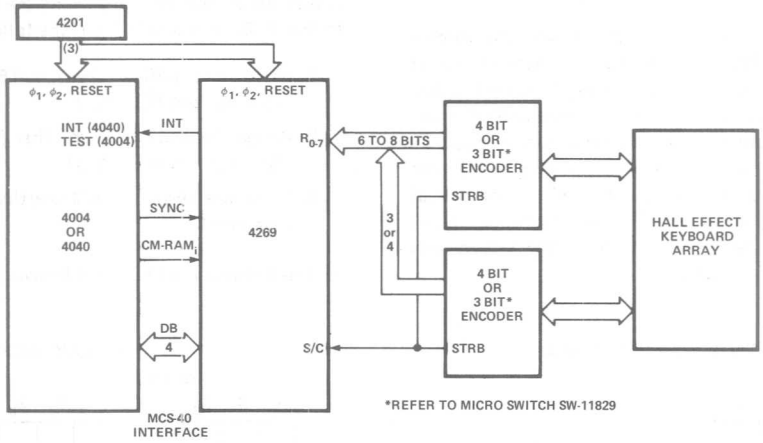


Figure 4. Strobed Keyboard Interface Encoded Keyboard Mode.

4269 Input Mode Interface to Sensor Matrix

The keyboard portion of the 4269 PKD can be set to Scanned Sensor mode to interface to a sensor matrix of up to 64 sensor points. The sensor points can be any type of electrical or mechanical sensor closures. Figure 5 illustrates a 4269 interfaced to an electrical sensor matrix.

Note that the Scanned Sensor mode may also be used as a general purpose 8-bit buffered input port. The internal keyboard sensor RAM can store up to 8 sets of 8-bit input data values.

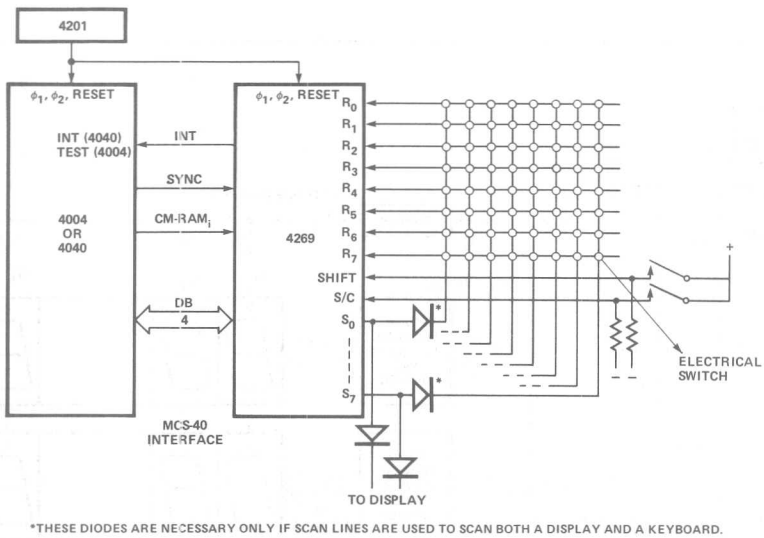


Figure 5. 64-Point Scanned Sensor Matrix.

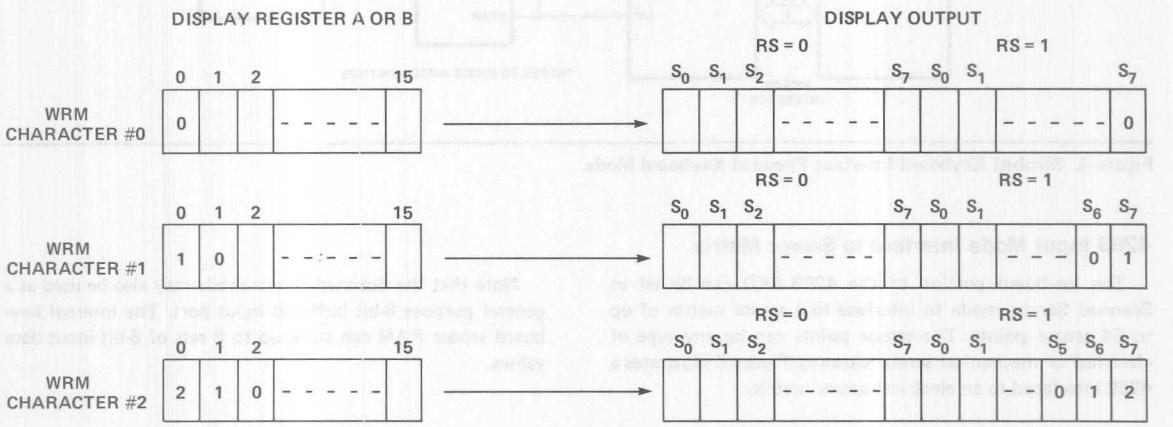
4269 Display Interface to Dual 8 x 4 Numeric Scanned Displays

In the Individual, Scanned Display mode, the display portion of the 4269 PKD can interface to a wide variety of display devices such as individual LEDs, 7-segment LEDs, phosphorescent displays, etc. Figure 6 illustrates a 4269 interface to dual 8 x 4 numeric scanned displays such as might be found in a cash register or a weighing scale. Figure 7 illustrates how the 4269 interface can be expanded to interface to dual 16 x 4 numeric scanned displays. The only additional circuit required is a demultiplexer to demultiplex the 8 Scan lines and the RS output of the 4269.

In such 16 character displays, if it is desired to have successive WRM instructions to shift the display from the LSD to the MSD as in a calculator, the following should be done:

1. Assign the LSD to be the 15th character scanned ($S_7 = V_{SS}$ and $RS = V_{SS}$);
2. Assign the MSD to be the first (#0) character scanned ($S_0 = V_{SS}$ and $RS = V_{DD}$);
3. Load the display RAM starting at Display Register character 0.

This technique is illustrated below:



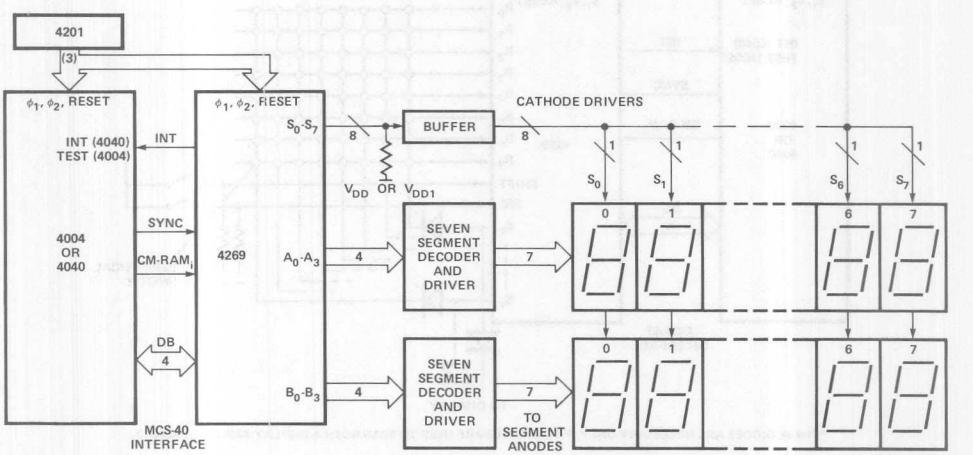


Figure 6. 4269 Interface to Dual 8 x 4 Numeric Scanned Displays.

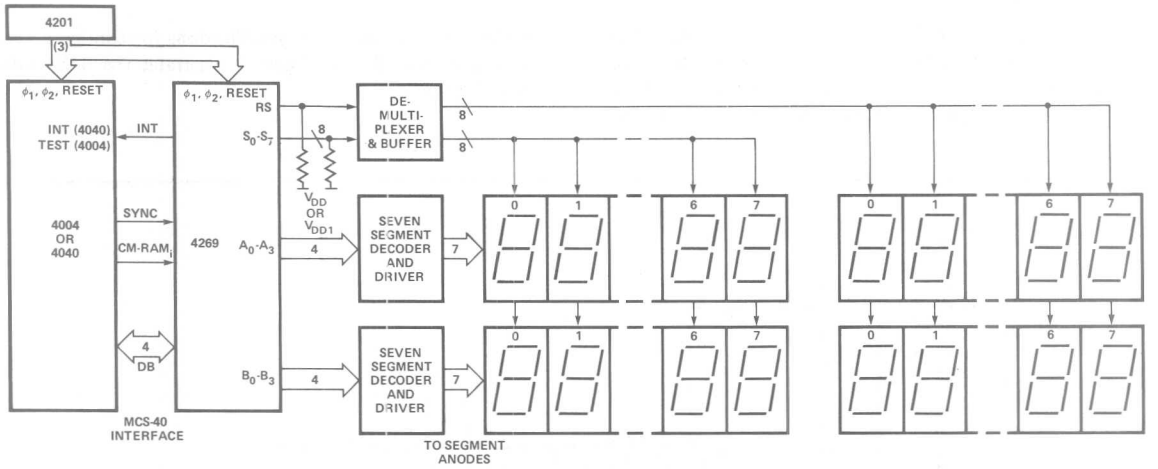


Figure 7. 4269 Interface to Dual 16 x 4 Numeric Scanned Displays.

4269 Display Interface to LED Array

Figure 8 illustrates a 4269 in the Individual, Scanned Display mode interfaced to an array of 64 LED lamps. Only transistor drivers are required between the 4269 A, B and Scan outputs and the LED lamps since these outputs are directly TTL compatible.

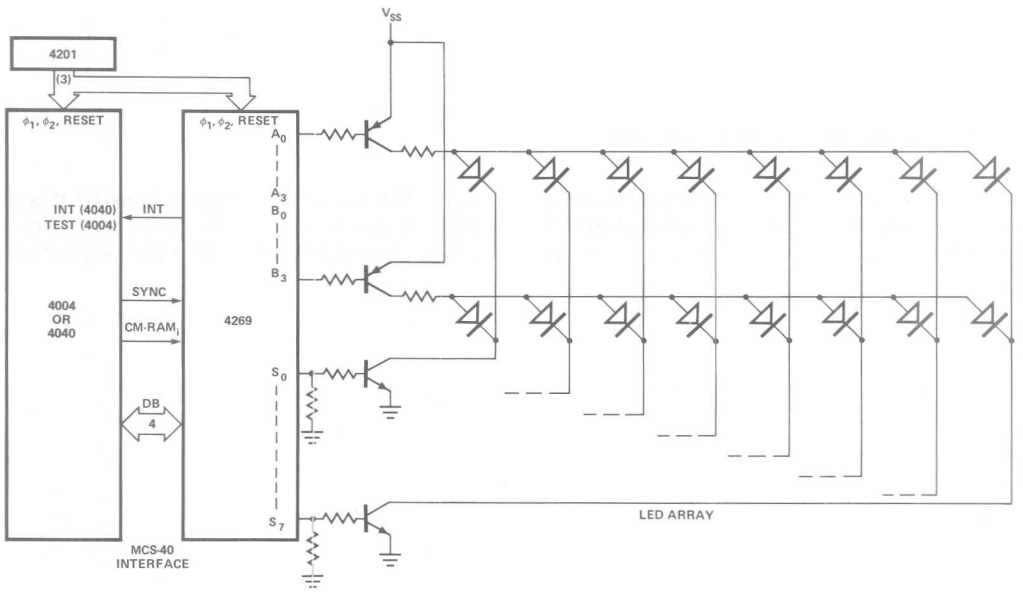


Figure 8. 4269 Interface to an Array of 64 LEDs.

4269 Display Interface to Burroughs Self-Scan® Display

The display portion of the 4269 provides a complete interface to Burroughs Self-Scan® devices in the 16, 18, or 20 character gas discharge modes. In these modes, the 4269 PKD generates the appropriate clock and reset timing signals required by the Self-Scan® specifications for continuous refreshing of the display. Figure 9 illustrates the 4269 Self-Scan® interface.

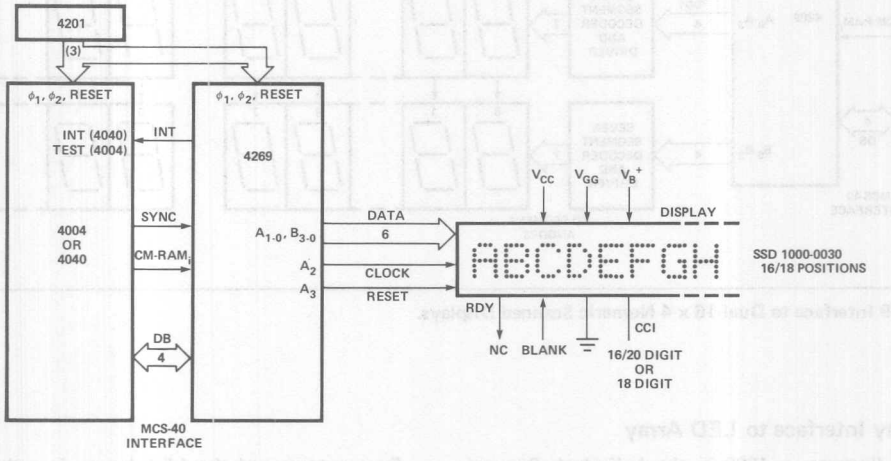


Figure 9. 4269 Interface to Burroughs Corp. Self Scan (16, 18 or 20 Character).

4269 Interface to Complete Keyboard/Display System

Figure 10 shows a system with a 4269 interfaced to both a keyboard and a display. In this particular system, a Panaplex II display (with Individual, Scanned Display mode set for 4269 PKD) and a 64-key scanned keyboard (Scanned Keyboard mode) are used. Note how the Scan lines of the 4269 are used to scan both the Panaplex display and the keyboard.

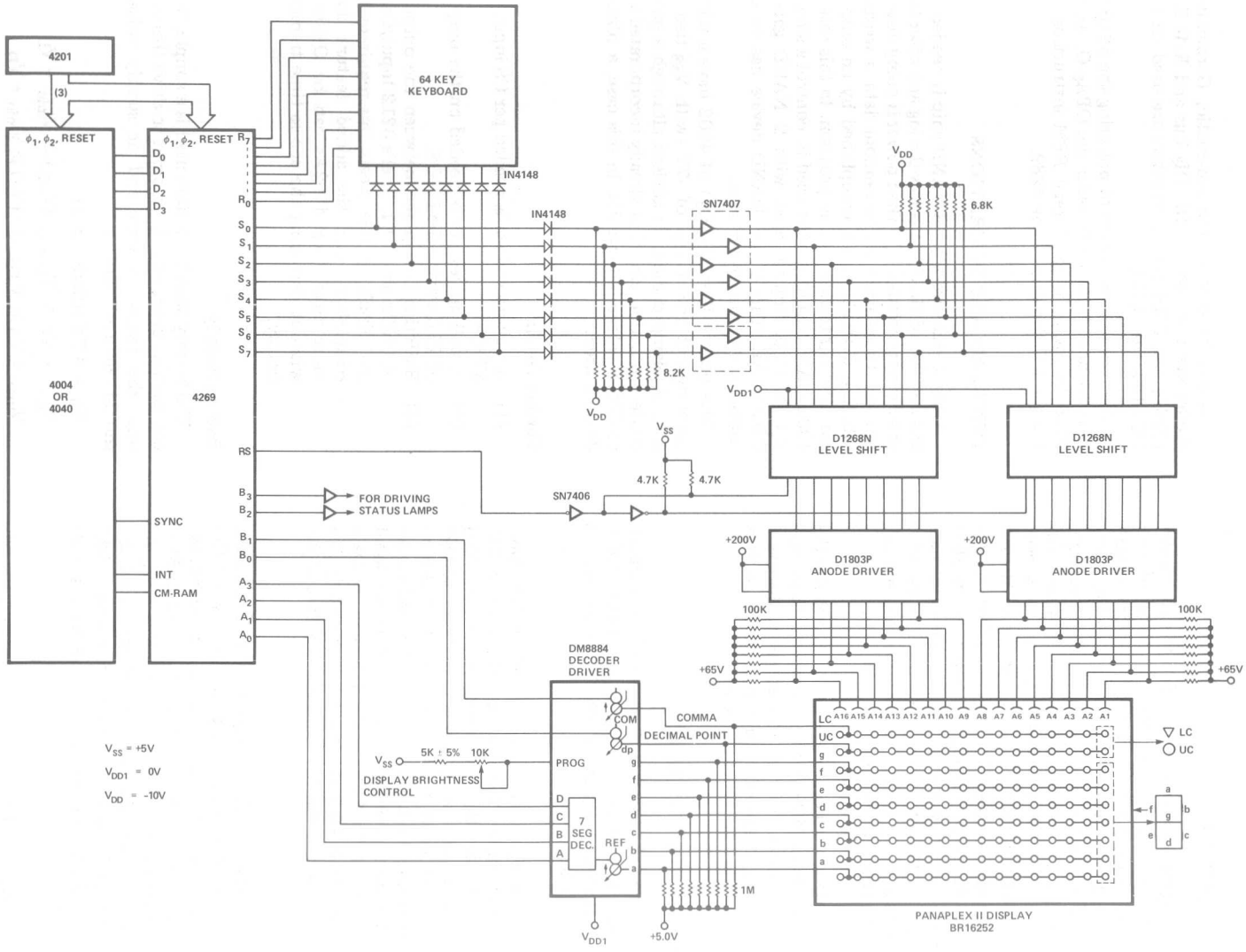


Figure 10. Complete Keyboard/Display System.

NON-VOLATILE MEMORY USING MCS-40™ WITH THE 5101 RAM

INTRODUCTION

The unpredictability of power failure in a volatile memory based system can result in a loss of irreplaceable information. Terminals, portable equipment and data collection instruments are but a few devices that require low cost non-volatile storage. Most read/write semiconductor memories are volatile i.e., information is lost when power is removed. Intel's 5101, 1K (256 x 4) CMOS static RAM with its extremely low standby power dissipation, typically 25μW, makes it feasible to retain information for weeks (444 days) using ordinary pen-light batteries on a "battery standby" mode. The use of a simple battery subsystem to maintain information can be a significant system cost reduction.

This note describes a technique for utilizing the 5101 RAM in an MCS-40 microcomputer based system. The specific MCS-40 configuration discussed here offers a means of maintaining processor data via batteries in a power standby mode.

4289 AND 5101 INTERFACE

The MCS-40 utilizes a 4289 standard memory interface chip to accommodate the 5101 RAM. The RAM CMOS and the 4289 PMOS necessitates family interface.

The Data Input lines (DI_0-DI_3) of the 5101 have a minimum input 'low' voltage (V_{IL}) of -0.3V, while the bi-directional I/O data lines (I/O_0-I/O_3) of the 4289 have a typical output 'low' voltage (V_{OL}) of -5V (with V_{SS} tied to +5V). With this incompatibility in voltages, buffers or clamped diodes (Germanium) are needed between the bi-directional I/O data lines (I/O_0-I/O_3) of the 4289 and the Data Input lines (DI_0-DI_3) of the 5101. This also applies to the PM line of the 4289 and the R/W line of the 5101.

The Data Output lines (DO_0-DO_3) of 5101 have a minimum output 'high' voltage of 2.4V, while the OPA_0-OPA_3 and OPR_0-OPR_3 need a minimum input 'high' voltage of 3.5V (with V_{SS} tied to 5V). Pull-up resistors are required on the OPR_0-OPR_3 of the 4289 to meet the required voltage. The DB_0-DB_3 lines of 3216 have a minimum output 'high' voltage of 3.5V, eliminating the need for any pull-up resistors.

With V_{DD1} of the 4289 tied to ground, the address and chip select lines are TTL compatible, eliminating the need for any buffering.

As can be seen in the schematic, Germanium diodes are used on the DI_0-DI_3 bus and R/W line (5101) and 3.3K pull-up resistors are used on the DO_0-DO_3 bus (5101).

The user has the option of not using the 3216 to channel data from 5101 onto the OPA_0-OPA_3 of 4289 by using an additional RPM instruction to flip the F/L flip-flop of the 4289.

INTERFACE CONSIDERATIONS

Only 1 standard CMOS NAND chip is needed to ensure CE2 of the 5101 is low during and after the process of power failure. When power is going down, one has to ensure that no random data is written into the 5101. This is accomplished by an output port and controlled by the program. In this case, a 4002 RAM output port is used to control a simple RS flip-flop, implemented with 2 NAND gates, CD4011AE. This CMOS NAND device has to be backed up by the battery also.

The output lines (O_0-O_3) of 4002 have a minimum output low voltage of -7V (with V_{SS} tied to 5V). A clamped diode is advised although a gate-oxide protection circuit is already incorporated into CMOS integrated circuits. In this case, a silicon diode is used.

Further Details

(1) A pull-up resistor is needed per CS input of 4702A.
(2) A pull-up resistor is needed on the output of TTL driving the CMOS.
(3) Buffering is required between the outputs of 4702A and 5101. Intel's 3212 Input/Output Bipolar device meets this requirement adequately, with the added feature that more than four 4702As can be OR-tied without degrading the access time tremendously.

Battery Supply

The battery standby system used is a simple, low cost parallel diode switch. In order to drive this system, the battery voltage and dc supply voltage should relate as follows:

V_D = 0.7V (diode drop)

$V_{max} + V_D \geq V_{battery}\ (V_{BB}) \geq V_{min} + V_D$

$V_{max} + V_D \geq V\ supply\ (V_S) \geq V_{min} + V_D$

Note: V_{max} and V_{min} refer to the 5101 and CD 4011AE.

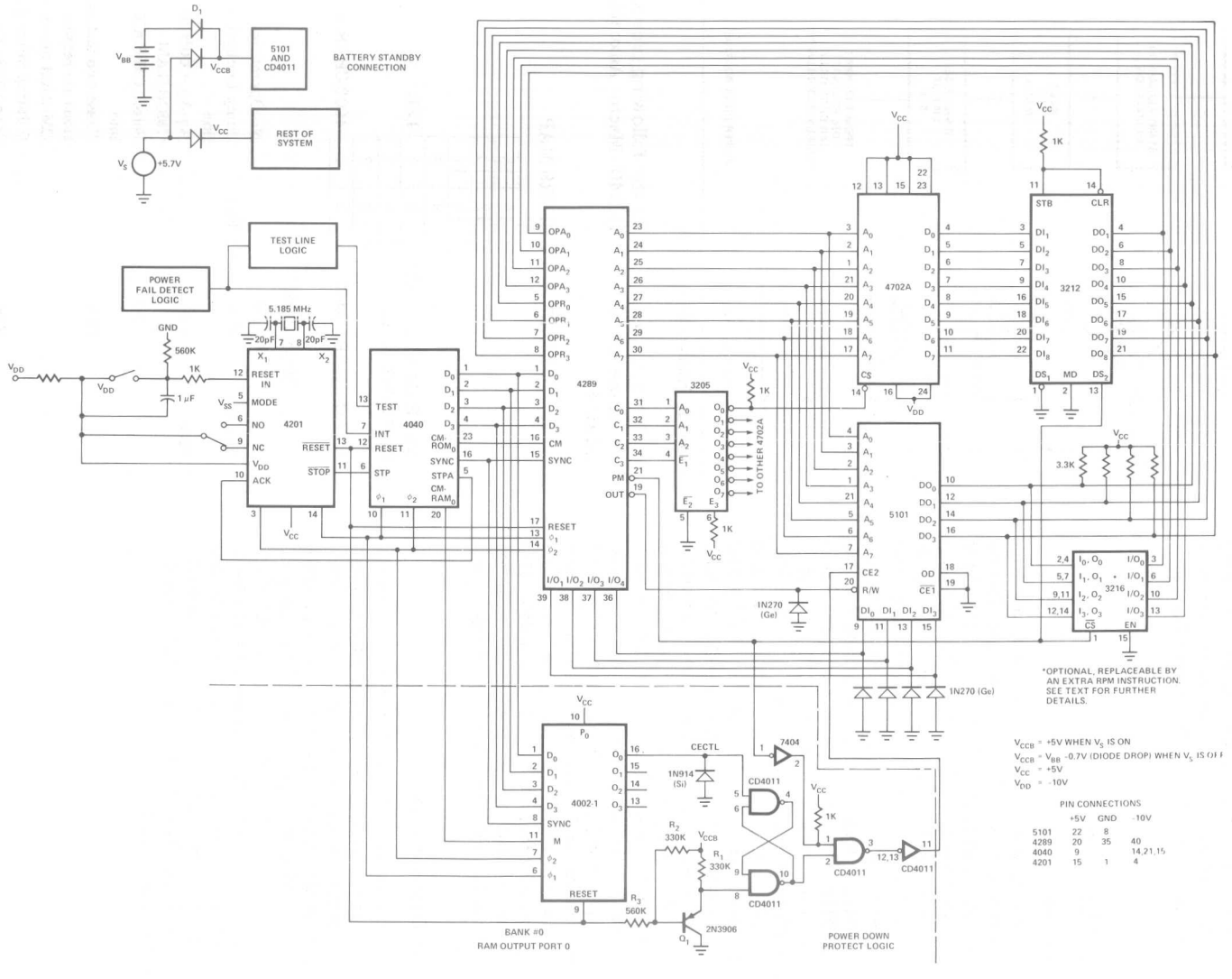


Figure 1. 4040 and 5101 Block Diagram.

In the event the supply drops below $V_{min.}$, the battery will forward bias diode D1 (refer to schematic) to form a closed-circuit and the 5101 and CD4011AE will continue to function properly through the battery. If a rechargeable battery is used, the battery can be trickle charged through a resistor.

Theory of Operation

Hardware Aspect

On detecting power up, the 4201 generates a reset pulse required by MCS-40 components. This reset pulse is also translated to TTL level by transistor Q_1 to enable operation of the 5101. The CPU has to be enabled for interrupt in order to recognize any interrupt. On detection of a power failure, the CPU is interrupted and 4040 begins program execution at memory location 3. Either a Power Down Routine (PDR) starts at location 3 or it contains a jump vector to the PDR. With one 5101, more than three 4002 memories can be saved in it. The F/L line of 4289 is not used in both writing into and reading from 5101. After the PDR, the RS flip-flop has to be toggled by the 4002 to disable the CE2 line so as to ensure that no random data is written into the 5101 during the power transitions. This is done by bringing the CECTL low (refer to schematic). All the above operations have to be done before the power supply drops below the minimum required voltage for the system. The time depends on how much memory one needs to save and what other I/O procedures need to be accomplished. This implies that the DC power supply must maintain power for a limited time after a line drop occurs.

Software Considerations

As the operation of the system depends entirely on the program, careful consideration must be given to the construction of the program given the limited time that the CPU has before the voltage drops below the minimum requirement. The following program is written only to save the majority of the 4002 (specifically, 4 registers of the 16 main memory characters — 64 characters). The test line is used to distinguish whether a power failure has occurred. Test line is false (0) when no power failure has occurred and true otherwise.

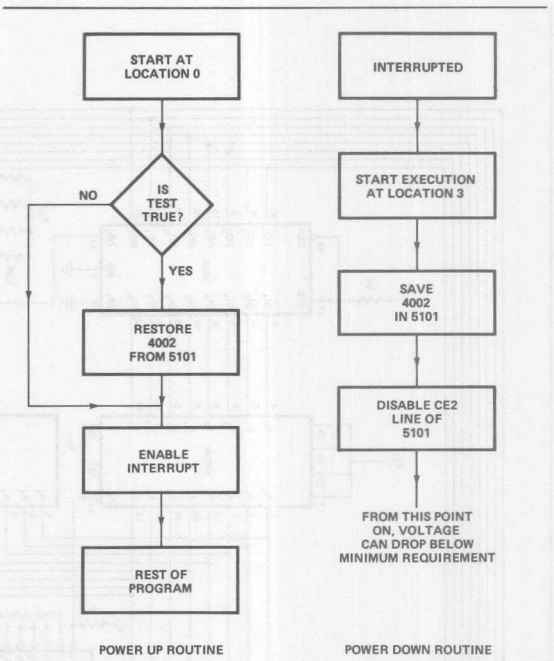


Figure 2. Program Flow Chart.

The mnemonics used in the following program are those of the 4004/4040 Macro Assembler (MAC 4).

INDEX REGISTERS MAP

		14	15	
		12	13	
		10	11	
		8	9	
5101 RAM ADR	MSB	6	7	LSB
RAM PORT ADR		4	5	
		2	3	
4002 CHIP/REG ADR		0	1	4002 CHAR ADR

MAIN PROGRAM

```
        NOP                 ;No Operation
START:  JUN   CKTEST        :Jump to check test
                            ;line
PDR:    FIM   6,0           ;A7-A0 = 00H for
                            ;CMOS RAM
        LDM   0             ;Select CM-RAM0
                            ;line
        DCL                 ;These two instruc-
                            ;tions are required if
                            ;CM-RAM line is not
                            ;0 during interrupt
        FIM   0,0           ;Save 4002s Reg. 0
                            ;in CMOS RAM
```

```
            JMS   SAVE
            INC   0          ;Save 4002s Reg. 1
            JMS   SAVE
            INC   0          ;Save 4002s Reg. 2
            JMS   SAVE
            INC   0          ;Save 4002s Reg. 3
            JMS   SAVE
            FIM   4,PORT 0   ;PORT 0=00000000B
            SRC   4          ;Set-up RAM port 0
            LDM   CECTL      ;CECTL=0001B cor-
                             ;responds to O0 line
            WMP              ;Disable CE2 line
HERE:       JUN   HERE       ;Wait for power to
                             ;go down.
CKTEST:     JNT   INIT
PUR:        FIM   6,0        ;A7-A0 = 00H
                             ;CM-RAM0 is auto-
                             ;matically selected
                             ;after reset so that
                             ;no LDM 0, DCL
                             ;needed
            FIM   0,0        ;Restore 4002s Reg.
                             ;0 from
            JMS   RESTORE    ;CMOS RAM
            INC   0          ;Restore 4002s
                             ;Reg. 1
            JMS   RESTORE
            INC   0          ;Restore 4002s
                             ;Reg 2
            JMS   RESTORE
            INC   0          ;Restore 4002s
                             ;Reg. 3
            JMS   RESTORE
INIT:       EIN              ;Enable interrupt
```

From this point on, normal processing can proceed.

SUBROUTINES

```
SAVE:       SRC   0          ;Set-up 4002 RAM
                             ;character
            RDM              ;Fetch RAM charac-
                             ;ter
            SRC   6          ;Set-up 5101s
                             ;address
            WPM              ;Write into CMOS
                             ;RAM. Note that
                             ;only one WPM is re-
                             ;quired because the
                             ;use of the 3216
                             ;avoids the need of
                             ;toggling the F/L
                             ;flip-flop
            INC   7          ;Increment 5101s
                             ;address A3-A0
            ISZ   1,SAVE     ;Point to next 4002s
                             ;RAM character and
                             ;continue until all 16
                             ;main characters are
                             ;saved
            INC   6          ;Increment 5101s ad-
                             ;dress A7-A4
            BBL   0          ;Return
RESTORE:    SRC   6          ;Set-up 5101s
                             ;address
            RPM              ;Fetch data from
                             ;5101
            SRC   0          ;Set-up 4002s RAM
                             ;character
            WRM              ;Restores it
            INC   7          ;Increment 5101s
                             ;address A3-A0
            ISZ   1,RESTORE  ;Point to next 4002s
                             ;RAM character and
                             ;continue until all 16
                             ;main characters are
                             ;restored
            INC   6          ;Increment 5101s
                             ;address A7-A4
            BBL   0          ;Return
```

The subroutine called SAVE is to save 4002s RAM characters into 5101. The data is saved sequentially starting at address 00. The above power down routine requires 478 memory cycles. With a 10.8 μs per memory cycle, the power supply has to maintain the minimum required voltage for at least 5.16 ms (478 x 10.8 μs = 5160.4 μs).

The subroutine called RESTORE is to restore 4002s RAM characters from 5101.

Note that CPU was not enabled for interrupt until after all the restoring was finished.

System Performance

The 2 CMOS chips, CD4011AE and 5101, draw a maximum of (15 + 15) μA = 30 μA, and Q1, R1, R2, R3 draw a maximum of 7.5 μA (with V_{CCB} = 4V). With a total of 37.5 μA on a standby mode, data retention can be maintained for 444 days using a 0.4 ampere-hour battery system. The 256 x 4 organization of the 5101 makes it suitable as a substitute for 4002 on standby mode.

The schematic shown can address up to 2K of ROM i.e., 8 of 4702As. R1, R2 and R3 can be optimized to draw less current depending on the transistor used.

Alternate System Configuration

The Power Down Protect Logic (PDPL) which comprises of 4002, CD4011AE and 7404 can be left out if the user does not require the power down protect capability. If PDPL were left out, the following connections have to be modified:

(1) Tie CE2 of 5101 high
(2) Connect PM of 4289 to $\overline{CE1}$ of 5101.

The 3216 can be left out if the user chooses to use an extra RPM and an extra WPM instruction to keep track of the F/L flip-flop of 4289. If the 3216 were left out, the RESTORE subroutine would then be the following.

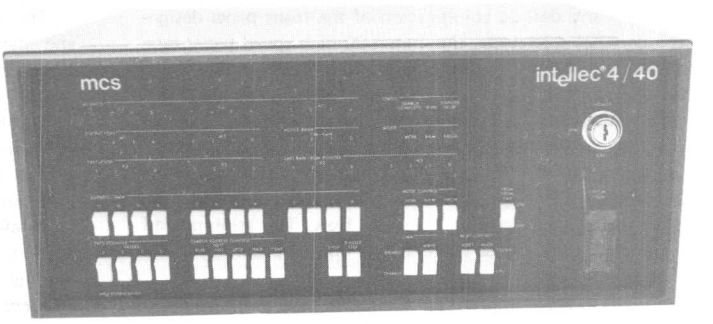
```
RESTORE:  SRC   6
          RPM               ;This is the dummy
                            ;instruction in
                            ;place of the 3216.
                            ;After reset, the first
                            ;RPM will read
                            ;OPA0-OPA3. Be-
                            ;cause the DO0-DO3
                            ;of 5101 are tied to
                            ;OPR0-OPR3, this
                            ;first RPM does not
                            ;pick up any useful
                            ;information. It
                            ;only serves to flip
                            ;the F/L flip-flop so
                            ;as to enable the
                            ;next RPM to ac-
                            ;cess the useful in-
                            ;formation.
          RPM               ;Fetch data from
                            ;5101
          SRC   0
          WRM
          INC   7
          ISZ   1,RESTORE
          INC   6
          BBL   0
```

Similarly, two WPM instructions would be required in the SAVE subroutine.

Conclusion

The 5101 as an MCS-40 Data Memory element can reduce the power consumption during the power down or standby mode. The use of low cost batteries to maintain important system data during a standby mode dramatically reduces user system cost over alternative methods. This is particularly true when small quantities of memory are involved.

CHAPTER 7. DEVELOPMENT AIDS



INTELLEC® 4 /MOD 40 MICROCOMPUTER DEVELOPMENT SYSTEM*

- **Complete hardware/software development system for the design and implementation of 4004 and 4040 CPU based microcomputer systems.**
- TTY interface, front panel designer's console, and an optional high speed paper tape reader interface, in conjunction with PROM resident system monitor provide complete program loading, punching, monitoring, interrogation, and alteration capabilities.
- **Program RAM (4K 8 bit bytes) provides a program development medium which lends itself to rapid program monitoring and alteration.**
- **Data RAM (320 4 bit nibbles expandable to 2560 nibbles) provides data storage capacity.**
- **Program PROM (expandable to 4K 8 bit bytes) in conjunction with the resident PROM programmer provide capability of simulating final ROM resident program.**
- **PROM resident system monitor and RAM resident macro-assembler included in standard systems software.**
- **Includes such standard program development features as program single step, address search (and pass count), next instruction indication, program flow verification.**
- **I/O expandable to 16 4 bit input ports and 48 4 bit output ports (all TTL compatible) allowing "hands-on" simulation of entire user system (processor and peripheral devices).**
- **RESET, STOP, INTERRUPT control signals available to user via back panel.**
- **Modular design with expansion capability provided for up to eleven optional or user designed modules.**

*Intellec is a registered trademark.

The Intellec 4/MOD 40 (imm 4-44A) system is a complete, self-contained microcomputer development system designed specifically to support the development and implementation of 4004 and 4040 CPU based microcomputer systems. Its modular design provides the flexibility to adapt to any size user system and the resident software greatly facilitates program development.

The basic Intellec 4/MOD 40 system consists of a 4 microcomputer modules (CPU, RAM, MEMORY CONTROL, and PROM PROGRAMMER), power supplies, I/O connectors, console, and displays. The heart of the system is the imm 4-43 central processor module built around Intel's high performance 4 bit 4040 CPU on a single chip. The imm 4-43 is a complete microcomputer system containing the system clock, 1K 8 bit bytes of PROM memory, 320 4 bit bytes of data RAM memory, 3 4 bit input ports and 8 4 bit output ports. The imm 6-28 program RAM memory module contains a 4K x 8 memory array composed of Intel 2102 static random access memory elements. The imm 4-72 control module contains the circuitry required to interface the central processor module to the program RAM module. The imm 6-76 PROM programmer module provides the capability of programming Intel 4702A PROMs in conjunction with the front panel PROM socket and system monitor. All I/O ports are TTL compatible and accessible from the back panel 37 pin connectors. The front panel designer's console provides a means of monitoring and controlling system operation.

The modular design of the Intellec allows great design system flexibility. Program PROM can be expanded to 4K 8 bit bytes using imm 6-26 or imm 4-22 optional modules. Data RAM can be expanded to 2560 4 bit bytes using imm 4-24 modules. I/O capability can be expanded to 16 4 bit input and 48 4 bit output ports using optional imm 4-60 modules. The universal prototype card (imm 6-70) in conjunction with the eleven optional card sockets (which contain all essential system signals) provide the capability for interfacing custom designed modules.

The user RESET, IN/OUT, STOP/STOP ACKNOWLEDGE, and INTERRUPT/INTERRUPT ACKNOWLEDGE control signals are all available at the back panel. Hence, the

user can interrupt, halt, and reset the resident CPU via his own interface.

Program interrogation and alteration can be accomplished by using any desired combination of the front panel designer's console, a teletype, the imm 4-90 high speed paper tape reader, and other Intellec compatible peripherals. The front panel designer's console provides the capability of manually writing data into memory and displaying memory contents, monitoring CPU bus contents during each processor subcycle, "freezing" system status after execution of a predefined instruction after a specified number of passes, single-stepping the program and verifying program flow. The teletype and reader serve as vehicles to input and output paper tapes and execute the system monitor.

Every Intellec 4/MOD 40 system comes with two systems software products – the PROM resident system monitor and the RAM resident assembler. The assembler has a paper-tape editor feature. The systems software is a powerful application program development tool.

The system monitor provides the capability of displaying and modifying memory contents, reading and punching object tapes, dynamically assigning system peripherals, programming and verifying PROMs, and performing other functions which significantly reduce program debug and development time.

The Intellec 4/MOD 40 RAM resident assembler translates source code into object code which will execute on the Intellec 4/MOD 40 or any MCS-40™ system. The assembler collects information from the source program, builds an internal symbol table, outputs a listing of the assembled program including error messages, and punches an object program tape.

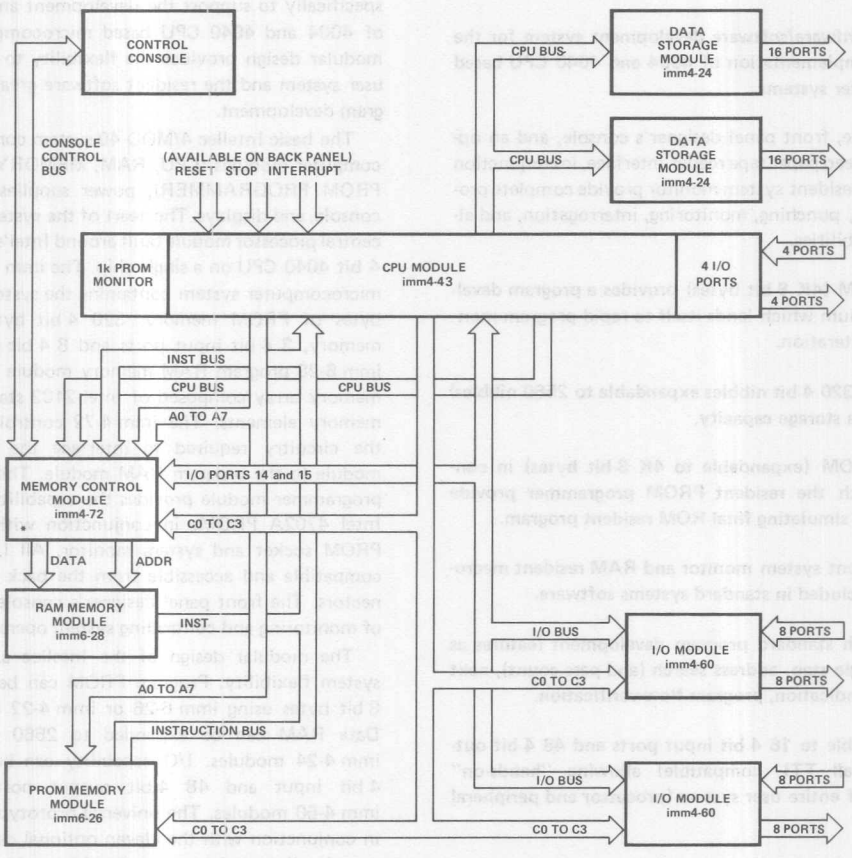


Figure 7-1. Intellec 4/Mod 40 System Block Diagram.

Specifications

Word Size

Data: 4 bits
Instructions: 8 bits/16 bits

Memory Size

5K bytes expandable to 12K bytes (combination of PROM, Data RAM, Program RAM) in three 4K byte memories selectable for execution from the front panel.

Instruction Set

60 instructions including conditionals, binary and decimal arithmetic, and I/O

Instruction Cycle Time

10.8 microseconds

System Clock

Crystal-controlled at nominal 5.185 MHz

I/O Channels

All ports are 4-line TTL
3 input ports expandable to 16
8 output ports expandable to 48

Interrupt

Available at back panel

Console Memory Access

Standard via control console

Memory Access Time

1 μs with standard memory modules

Environmental Characteristics

Operating temperature: 0°C to 55°C

Electrical Characteristics

DC power supplies: V_{CC} = 5V ± 5%
I_{CC} = 12A
V_{DD} = -10V ± 5%
I_{DD} = 1.8A

AC power supplies: Mod 40: 60 Hz, 115 VAC @ 200 W
Mod 40/220: 50 Hz, 230 VAC @ 200 W

Physical Characteristics

Intellec 4/40: 7" x 17 1/8" x 12 1/4" (table top only; optional rack mount available)
Weight: 30 lb. (13.61 kg.)

Optional Modules

Available for the Intellec 4/MOD 40:

- imm 4-22 Instruction/Data Storage Module
- imm 4-24 Data Storage Module
- imm 4-60 Input/Output Module*
- imm 6-26 PROM Memory Module
- imm 6-28 RAM Memory Modules (Additional)
- imm 6-36 Rack Mounting Kit
- imm 6-70 Universal Prototype Module
- imm 6-72 Module Extender
- imm 4-90 High Speed Paper Tape Reader*

Equipment Supplied

Central Processor Module
RAM Memory Module
PROM Programmer Module
Memory Control Module
Chassis with Mother Board
Power Supplies
Control and Display Panel
Finished Cabinet
PROM Resident System Monitor
RAM Resident Assembler

Complete Hardware and Software Documentation

Programmers Manual
Operators Manual
Hardware Reference Manual
Module Schematics

*The imm 4-60 Input/Output Module is necessary if the imm 4-90 High Speed Paper Tape Reader is to be used.

MCS-40™ SOFTWARE SYSTEM DEVELOPMENT TOOLS

Programming for the MCS-40 microcomputer can be done easily and quickly using Intel's cross macro assembler, MAC4. This powerful assembler translates three letter mnemonics representing each MCS-40 instruction into a numeric representation that may be loaded directly into an Intellec 4 development system or programmed to ROM. Advanced MAC4 features provide full macro capability and conditional assembly capability. Output is in hexadecimal or BNPF code.

MAC4 is written in ANSI standard FORTRAN IV and is designed to run on any large scale computer system (32 bit word size or larger) with little or no modification. The FORTRAN source program for MAC4 is available on magnetic tape directly from Intel. In addition, MAC4 may be used on either TYMSHARE, UNITED COMPUTING SYSTEMS, or GENERAL ELECTRIC worldwide timesharing services and may also be used on TIMESHARING LTD in Europe. Contact these services directly for further information.

A simulator program called INTERP/40 is also available. This program provides a complete software simulation of MCS-40 programs. INTERP/40 is written in ANSI standard FORTRAN IV and thus allows the designer to debug his software interactively on any large computer system.

MCS-40™ USER'S LIBRARY

The MCS-40 User's Library is a collection of programs written by users of the 4004 and 4040 CPU chips. These programs have been contributed to the user's library for the benefit of all MCS-40 users. Intel will make source listings of all programs and detailed instructions on their use available to all members of the MCS-40 User's Library. To become a member simply:

1. Submit a program to the library with detailed documentation and a completed user's library submittal form, or
2. Pay a yearly membership fee.

For more information, contact your local Intel representative. Some of the current programs in the library are listed below.

4-BIT USERS LIBRARY

Title	Function
CROSS PROGRAMS	
MCS-40 Cross Assembler for Intellec MDS	Program to perform assembly of 4004/4040 programs on an Intellec MDS.
MCS-40 Cross Assembler for Intellec 8/Mod 80	Program to perform assembly of 4004/4040 programs on an Intellec 8/Mod 80.
Intel MCS-40 Cross Assembler and Text Editor	Edit and assemble MCS-40 source language using a Computer Automation Alpha-16/Alpha-LSI producing program listings, error diagnostics, source and object tapes.
Cross Assembler on PDP-8	Performs symbolic assembly for 4004 assembly language programs. The assembler runs on a DEC PDP-8 minicomputer.
Cross Assembler for NOVA Computer	Assemble 4004 programs and program PROMs.
BNPF Tape Generator for PDP-8	Produces a BNPF object tape from the output of the PDP-8 assembler modified to assemble 4004 programs.
MCS-4 Simulator for PDP-8	Simulates operation of 4004 system; allows breakpoints, dumps, modification of RAM or ROM, I/O under user control, provides cycle counter for timing. Accepts output of PAL8 assembler in binary form.
MCS-4 Simulator for NOVA	Simulates operation of the 4004 on a NOVA (16K).
PROGRAMS FOR 4004 AND 4040 CPUs	
I/O Programs	
I/O Test	To exercise all I/O lines to allow for trouble-shooting of system design, wiring errors, and chip malfunctions.
High Speed Printer Interface	Subroutine to write the ASCII character stored in IR2 and IR3 to a Centronics 101 high speed printer.
TP3100	This is a subroutine which provides a software interface to a Bowmar TP3100 thermal printer. The subroutine provides all timing, as well as control and character print functions.
Bowmar TP3100 Printer Routine	This program is to run a Bowmar model 3100 thermal printer.
Peripheral Interface Routine for a Thermal Strip Printer	This subroutine controls the printing of data (numbers and selected characters) on thermal print paper using a 4 x 5 dot matrix thermal printhead. The software provides character generation and controls the timing of the print cycle.

4 x 8 Keyboard Scanner	Subroutine to scan, read, debounce and store the row and column identification of an actuated (normally open) switch wired into a 4 x 8 matrix.
Subroutine to Scan a TOUCHTONE® Keyboard	Scans 3 x 4 switch array of TOUCHTONE® keyboard.
8-Digit Register Display	Program to display 8 digits of data.
IOMEC SERIES THREE (S-3) Cartritape to Intel MCS-40	Routines which allow full control of the IOMEC S-3 by a 4004 or 4040.
Paper Tape Programs	
Paper Tape Edit	Add, correct, or delete lines in generating a new paper tape without manually controlling tape reader.
Paper Tape Conversion	This program converts information originally on 5-level (Telex and TWX) paper tape to ASCII 8-level paper tape. In this way, programs can be sent over Telex or TWX lines and then converted without retyping the information.
Development Support Programs	
Intellec 4/Mod 40 - Silent 700 Interface	Program to interface Intellec 4/Mod 40 to TI Silent 700 terminal.
PROM Dump Utility Program	This is a program to dump the contents of a PROM in the front panel socket onto the teletype printer. The first address and first word is always printed out in the form "00-00" as address-contents. All subsequent address-contents listings are printed out only if the contents of the respective location are different from the contents of the previous location.
Pro Forma	This program assists in the compiling of source code tapes by eliminating errors and typing mistakes. In the keyboard mode it will only transmit characters to the paper tape punch that are valid in the context of the system assembly language, and automatically formats individual lines and pages to suit.
MCS-40 Dissembler	To convert 4004 or 4040 machine code programs back into mnemonic or assembly code to assist in the modification of programs.
Delay Programs	
Delay Subroutines	To conditionally generate a selectable time delay of: 1. 1 through 256 ms in one ms increments 2. 1 through 256 quarterseconds in one quartersecond increments 3. 1 through 15 minutes in one minute increments
Bit Manipulation Programs	
Bit Manipulation Routine	AND, OR, XOR, COMPARE, set unselected bits, clear selected bits, test ones.
Universal Logic Subroutines	Forms logical AND, OR, XOR, $\overline{\text{XOR}}$ functions between the contents of Registers 0 and 1.
8-Bit Parity Check Annex	Compute parity of 8-bit word without affecting any registers or carry.
Parity Checker/Generator	Routine to check or generate parity for 8-bit byte. Utilizes modulo-2 counting technique.
Parity Generator, ASCII Character	Routine to add even parity bits to 7-bit ASCII character. Utilizes modulo-2 couting technique.
Code Conversion Programs	
ASCII to EBCDIC	Table and routine to perform 7-bit ASCII to 8-bit EBCDIC code conversion. Full table with 128 entries provided.
Binary to BCD Converter	Converts an 8-bit binary number to a BCD number.
4-Digit BCD to Binary Converter	Subroutine to convert a 4-digit binary coded decimal number into its straight binary equivalent.
HEXBCD	Convert 2 digit HEX value to decimal range.

Arithmetic Programs

4-Digit Multiply	Multiplies 4 digits by 4 digits to produce an 8-digit result. Result area is not automatically cleared so that partial sums of successive products are easily accumulated.
Multiply/Divide 8 Decimal Numbers	The multiply/divide subroutine calculates the product/quotient by repeated shifted additions or subtractions and by incrementing/decrementing the multiplier/quotient.
Fast Binary Multiply: Selectable Bit Precision and Constant Execute Time	The user loads the input variables to CPU registers and specifies one of five multiply precisions (12 x 12, 12 x 8, 8 x 8, 8 x 4, 4 x 4) via a code character in register E.
Fast Decimal Multiplication Routine	This subroutine computes the product (16 digit maximum) of two fixed point decimal numbers (each 8 digits maximum).
Floating Point Arithmetic Subroutine Package	Performs decimal arithmetic on 16-digit floating point operands (hexadecimal arithmetic is possible with minor changes). Numbers may range from 10^{-130} to 10^{125}. Functions include Addition, Subtraction, Multiplication, Division and a normalization routine.
A Chebyshev Approximation Package	The package contains approximation routines for sine, cosine, arctangent, natural logrithm ($\log_e$), and exponential functions (e^x). It also contains routines for performing addition, complement, multiplication, and division on 65-bit binary numbers.
Mobile Mean Program	Calculates the mean between the current value of a 3-digit BCD number and the previous four values.
Right Justified HEX Data Shifter	Shifts HEX digit (four binary bits) into RAM right justified.
Random Number Generator	Subroutine to generate a pseudo-random sequence of numbers.

Miscellaneous Programs

General Purpose ROM	General purpose I/O handler and debugger.
Data Compare	Compares two 8-digit numbers and returns a pointer to the greater value in the carry.
Selector Subroutine	To recognize 5 individual character sequences up to four characters each and set appropriate flags.
Automatic Digital Integration	Program will detect and integrate peaks from an amino acid analyzer and type out the area under the peak on the teletype. Program will detect saddle peaks and do simple baseline correction.
Solitaire Game of "LIFE"	Computes and displays successive generations of game area status. Accepts co-ordinates of points for the definition of initial conditions or for altering patterns between generation cycles.

INSTRUCTIONS FOR PROGRAM SUBMITTAL TO MCS USER'S LIBRARY

1. Complete Submittal Form as follows: (Please print or type)
   a. Processor (check appropriate box)
   b. Program title: Name or brief description of program function
   c. Function: Detailed description of operations performed by the program
   d. Required hardware:
      For example: TTY on port 0 and 1
      Interrupt circuitry
      I/O Interface
      Machine line and configuration for cross products
   e. Required software:
      For example: TTY routine
      Floating point package
      Support software required for cross products
   f. Input parameters: Description of register values, memory areas or values accepted from input ports
   g. Output results: Values to be expected in registers, memory areas or on output ports
   h. Program details (for resident products only)
      1. Registers modified
      2. RAM required (bytes)
      3. ROM required (bytes)
      4. Maximum subroutine nesting level
   i. Assembler/Compiler Used:
      For example: PL/M
      Intellec 8 Macro Assembler
      IBM 370 Fortran IV
   j. Programmer, company and address

2. A source listing of the program must be included. This should be the output listing of a compile or assembly, Extra information such as symbol table or code dumps is not necessary.

3. A test program which assures the validity of the contributed program must be included. This is for the user's verification after he has transcribed and assembled the program in question.

Your program will be photo-copied for publication in the User's Library. Please send an original, clear, un-marked copy.

Send completed documentation to:

Intel Corporation
User's Library
Microcomputer Systems
3065 Bowers Avenue
Santa Clara, California 95051

MICROCOMPUTER USER'S LIBRARY SUBMITTAL FORM

☐ 4004 ☐ 4040 ☐ 8008 ☐ 8080

(use additional sheets if necessary)

Program Title	
Function	
Required Hardware	
Required Software	
Input Parameters	
Output Results	

Registers Modified:	Assembler/Compiler Used:
RAM Required:	Programmer:
ROM Required:	Company:
Maximum Subroutine Nesting Level:	Address:

98-034B

MCS-40™ MICROCOMPUTER WORKSHOP

Intel offers regularly scheduled Microcomputer workshops on each of its Microcomputer product lines. In particular, the MCS-40 workshops cover materials necessary to the design and development of systems utilizing the MCS-40 components.

For current schedules and a copy of the course synopsis, contact your local Intel representative.

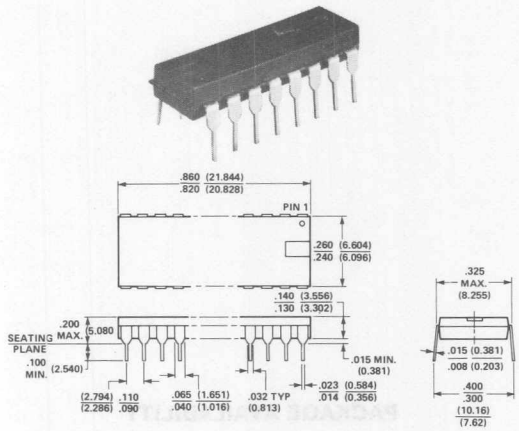
CHAPTER 8
PACKAGING AND ORDERING INFORMATION

PACKAGE AVAILABILITY

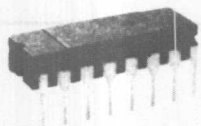
	Intel Product Number	Standard Package Type			Number of Pins	Comments
CPUs	4004	C	D		16	
	4040	C	D	P	24	
I/O	4003	C	D	P	16	
	4265		D	P	28	
	4269		D	P	40	
	8251	C		P	28	
	8253		D	P	24	Available 2nd Quarter, 1976
Peripherals	4201	C	D	P	16	
	4008	C	D	P	24	
	4009	C	D	P	24	
	4289	C	D	P	40	
	3205	C	D	P	16	
	3214	C	D	P	24	
	3216		D	P	16	
	3226		D	P	16	
RAMS	4002	C	D	P	16	
	4101	C		P	22	
	5101, 5101L	C		P	22	
PROMS AND ROMS	4001	C	D	P	16	
	4308	C		P	28	
	4316A	C		P	24	
	4702A	C			24	

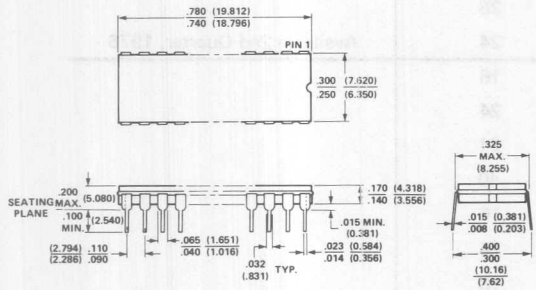
Type C = Hermetic
Type D = Hermetic
Type P = Plastic

16-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

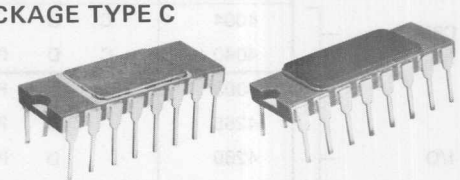


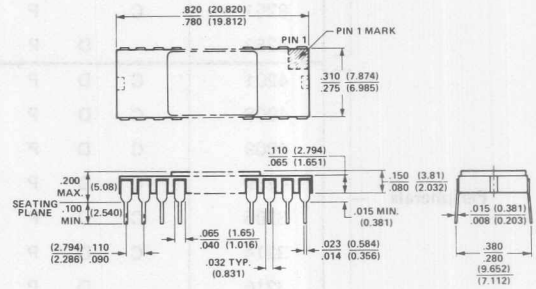
16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



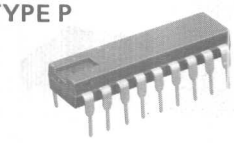


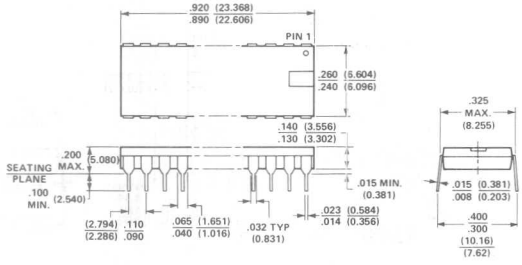
16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C



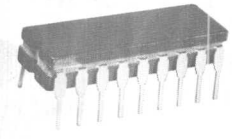


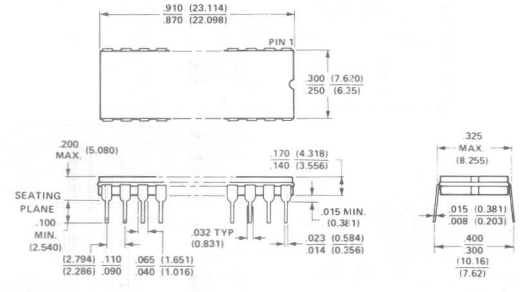
18-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



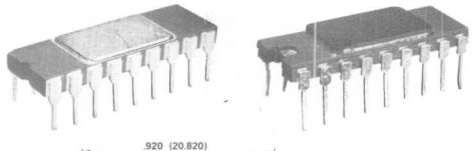


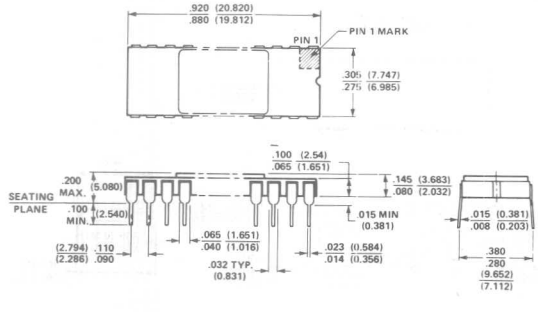
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



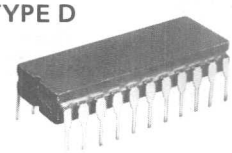


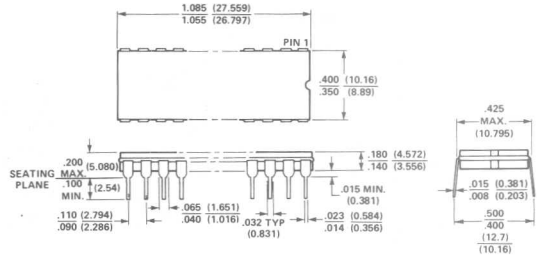
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C



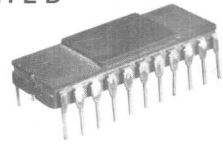


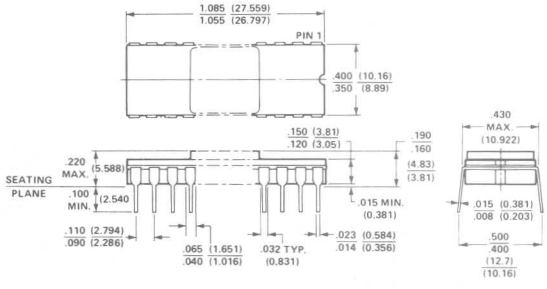
22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



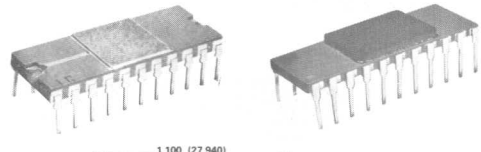


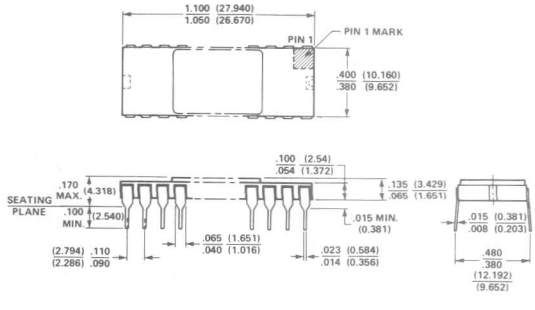
22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B



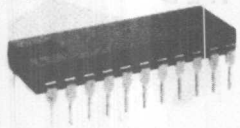


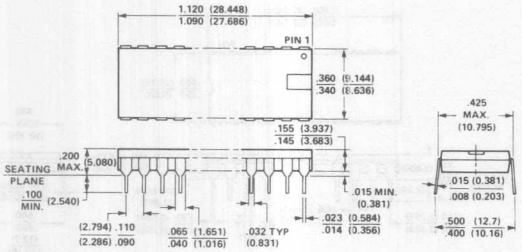
22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C



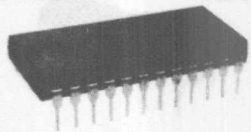


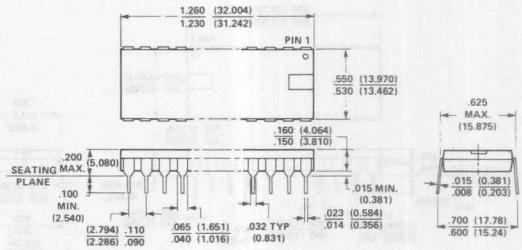
22-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P





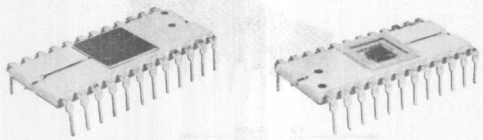
24-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

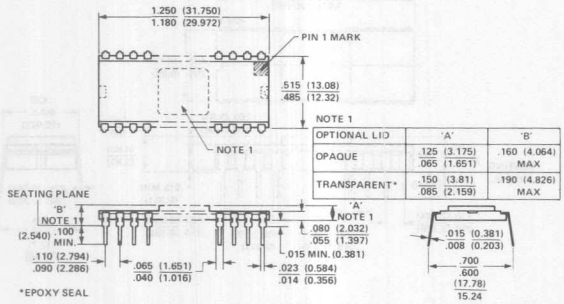




24-LEAD HERMETIC DUAL IN-LINE PACKAGE

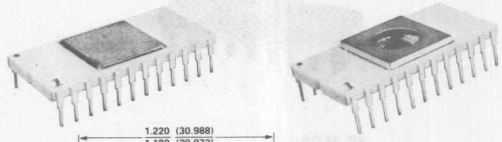
TYPE C

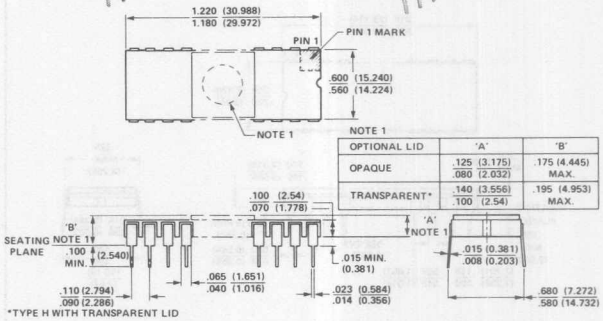




OPTIONAL LID	'A'	'B'
OPAQUE	.125 (3.175) .065 (1.651)	.160 (4.064) MAX.
TRANSPARENT*	.150 (3.81) .085 (2.159)	.190 (4.826) MAX

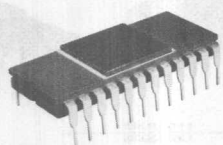
TYPE C OR H*

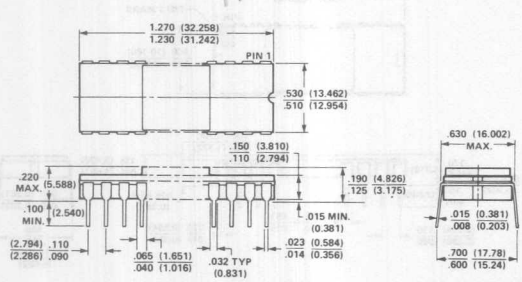




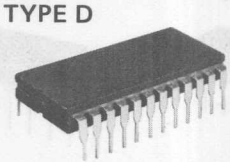
OPTIONAL LID	'A'	'B'
OPAQUE	.125 (3.175) .080 (2.032)	.175 (4.445) MAX.
TRANSPARENT*	.140 (3.556) .100 (2.54)	.195 (4.953) MAX.

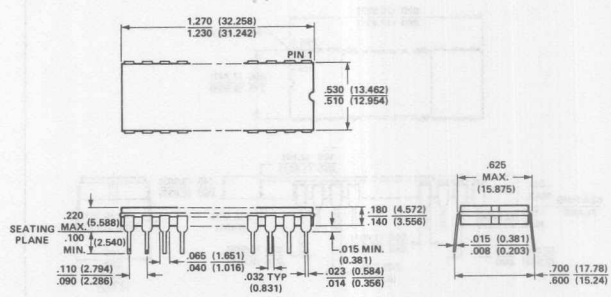
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE B





24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

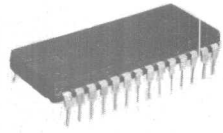


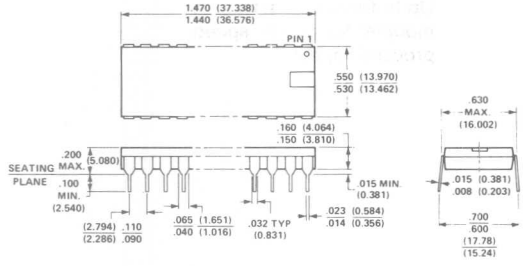


PACKAGING INFORMATION

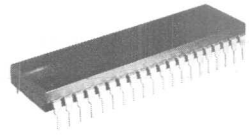
Dimensions in inches and (millimeters).

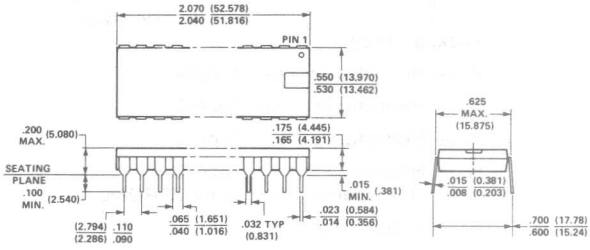
28-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



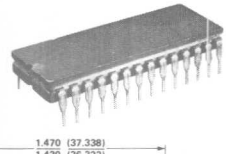


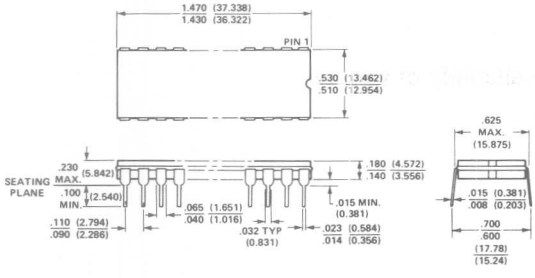
40-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



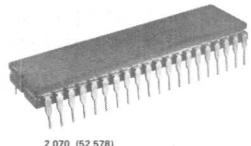


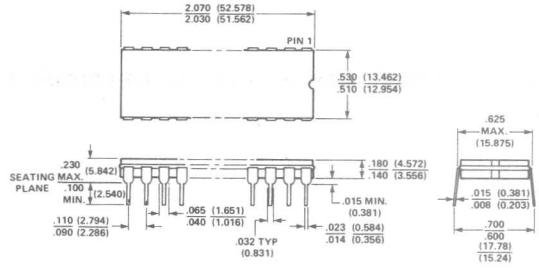
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



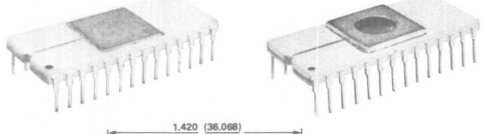


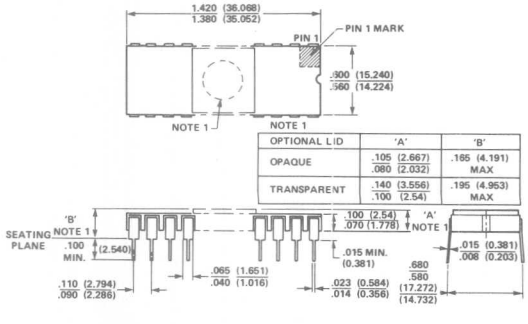
40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D





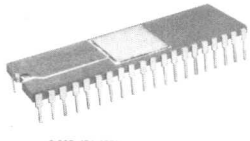
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

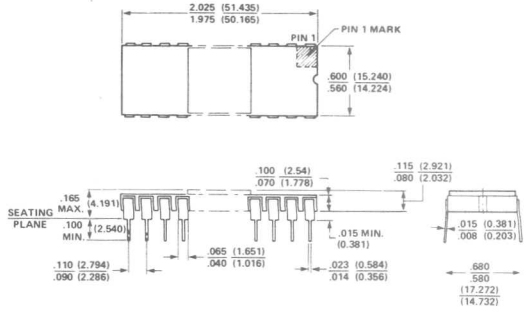




OPTIONAL LID	'A'	'B'
OPAQUE	.105 (2.667) .080 (2.032)	.165 (4.191) MAX
TRANSPARENT	.140 (3.556) .100 (2.54)	.195 (4.953) MAX

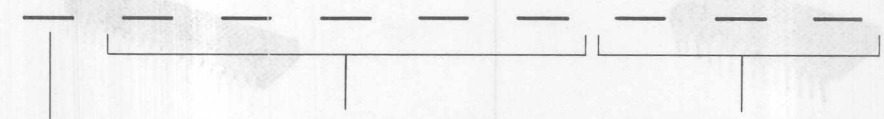
40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C





ORDERING INFORMATION

Semiconductor components are ordered as follows:



Four or five digit device type

Up to three character modifier for power, speed, processing, etc.

Package Type

B — Hermetic Package, Type B

C — Hermetic Package, Type C

D — Hermetic Package, Type D

H — Hermetic Package, Type C with Transparent Window for EPROMs

P — Plastic Package

Examples

P5101L	CMOS 256 x 4 RAM, low power selection, plastic package
C4040	4040 Microcomputer, hermetic package Type C

The latest Intel price book should be consulted for availability of various options.

MCS-40 Instruction Machine Codes

Hex	Mnemonic		Hex	Mnemonic			Hex	Mnemonic		Hex	Mnemonic	
00	NOP		40	JUN			80	ADD	0	C0	BBL	0
01	HLT		41	JUN			81	ADD	1	C1	BBL	1
02	BBS		42	JUN			82	ADD	2	C2	BBL	2
03	LCR		43	JUN			83	ADD	3	C3	BBL	3
04	OR4		44	JUN			84	ADD	4	C4	BBL	4
05	OR5		45	JUN			85	ADD	5	C5	BBL	5
06	AN6		46	JUN			86	ADD	6	C6	BBL	6
07	AN7	4040 Only	47	JUN			87	ADD	7	C7	BBL	7
08	DB0		48	JUN			88	ADD	8	C8	BBL	8
09	DB1		49	JUN			89	ADD	9	C9	BBL	9
0A	SB0		4A	JUN			8A	ADD	10	CA	BBL	10
0B	SB1		4B	JUN			8B	ADD	11	CB	BBL	11
0C	EIN		4C	JUN			8C	ADD	12	CC	BBL	12
0D	DIN		4D	JUN			8D	ADD	13	CD	BBL	13
0E	RPM		4E	JUN		Second hex	8E	ADD	14	CE	BBL	14
0F	–		4F	JUN		digit is part	8F	ADD	15	CF	BBL	15
10	JCN	CN=0	50	JMS		of jump	90	SUB	0	D0	LDM	0
11	JCN	CN=1 also JNT	51	JMS		address.	91	SUB	1	D1	LDM	1
12	JCN	CN=2 also JC	52	JMS			92	SUB	2	D2	LDM	2
13	JCN	CN=3	53	JMS			93	SUB	3	D3	LDM	3
14	JCN	CN=4 also JZ	54	JMS			94	SUB	4	D4	LDM	4
15	JCN	CN=5	55	JMS			95	SUB	5	D5	LDM	5
16	JCN	CN=6	56	JMS			96	SUB	6	D6	LDM	6
17	JCN	CN=7	57	JMS			97	SUB	7	D7	LDM	7
18	JCN	CN=8	58	JMS			98	SUB	8	D8	LDM	8
19	JCN	CN=9 also JT	59	JMS			99	SUB	9	D9	LDM	9
1A	JCN	CN=10 also JNC	5A	JMS			9A	SUB	10	DA	LDM	10
1B	JCN	CN=11	5B	JMS			9B	SUB	11	DB	LDM	11
1C	JCN	CN=12 also JNZ	5C	JMS			9C	SUB	12	DC	LDM	12
1D	JCN	CN=13	5D	JMS			9D	SUB	13	DD	LDM	13
1E	JCN	CN=14	5E	JMS			9E	SUB	14	DE	LDM	14
1F	JCN	CN=15	5F	JMS			9F	SUB	15	DF	LDM	15
20	FIM	0	60	INC	0		A0	LD	0	E0	WRM	
21	SRC	0	61	INC	1		A1	LD	1	E1	WMP	
22	FIM	2	62	INC	2		A2	LD	2	E2	WRR	
23	SRC	2	63	INC	3		A3	LD	3	E3	WPM	
24	FIM	4	64	INC	4		A4	LD	4	E4	WR0	
25	SRC	4	65	INC	5		A5	LD	5	E5	WR1	
26	FIM	6	66	INC	6		A6	LD	6	E6	WR2	
27	SRC	6	67	INC	7		A7	LD	7	E7	WR3	
28	FIM	8	68	INC	8		A8	LD	8	E8	SBM	
29	SRC	8	69	INC	9		A9	LD	9	E9	RDM	
2A	FIM	10	6A	INC	10		AA	LD	10	EA	RDR	
2B	SRC	10	6B	INC	11		AB	LD	11	EB	ADM	
2C	FIM	12	6C	INC	12		AC	LD	12	EC	RD0	
2D	SRC	12	6D	INC	13		AD	LD	13	ED	RD1	
2E	FIM	14	6E	INC	14		AE	LD	14	EE	RD2	
2F	SRC	14	6F	INC	15		AF	LD	15	EF	RD3	
30	FIN	0	70	ISZ	0		B0	XCH	0	F0	CLB	
31	JIN	0	71	ISZ	1		B1	XCH	1	F1	CLC	
32	FIN	2	72	ISZ	2		B2	XCH	2	F2	IAC	
33	JIN	2	73	ISZ	3		B3	XCH	3	F3	CMC	
34	FIN	4	74	ISZ	4		B4	XCH	4	F4	CMA	
35	JIN	4	75	ISZ	5		B5	XCH	5	F5	RAL	
36	FIN	6	76	ISZ	6		B6	XCH	6	F6	RAR	
37	JIN	6	77	ISZ	7		B7	XCH	7	F7	TCC	
38	FIN	8	78	ISZ	8		B8	XCH	8	F8	DAC	
39	JIN	8	79	ISZ	9		B9	XCH	9	F9	TCS	
3A	FIN	10	7A	ISZ	10		BA	XCH	10	FA	STC	
3B	JIN	10	7B	ISZ	11		BB	XCH	11	FB	DAA	
3C	FIN	12	7C	ISZ	12		BC	XCH	12	FC	KBP	
3D	JIN	12	7D	ISZ	13		BD	XCH	13	FD	DCL	
3E	FIN	14	7E	ISZ	14		BE	XCH	14	FE	–	
3F	JIN	14	7F	ISZ	15		BF	XCH	15	FF	–	

NOTES:

MCS-40™ Instruction Set

BASIC INSTRUCTIONS (* = 2 Word Instructions)

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
00	NOP	0 0 0 0	0 0 0 0	No operation.
1 -	*JCN	0 0 0 1	$C_1 C_2 C_3 C_4$	Jump to ROM address $A_2 A_2 A_2 A_2$, $A_1 A_1 A_1 A_1$ (within the same ROM that contains this JCN instruction) if condition $C_1 C_2 C_3 C_4$ is true, otherwise go to the next instruction in sequence.
- -		$A_2 A_2 A_2 A_2$	$A_1 A_1 A_1 A_1$	
2 -	*FIM	0 0 1 0	R R R 0	Fetch immediate (direct) from ROM Data $D_2 D_2 D_2 D_2 D_1 D_1 D_1 D_1$ to index register pair location RRR.
- -		$D_2 D_2 D_2 D_2$	$D_1 D_1 D_1 D_1$	
3 -	FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
3 -	JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A_1 and A_2 time in the instruction cycle.
4 -	*JUN	0 1 0 0	$A_3 A_3 A_3 A_3$	Jump unconditional to ROM address $A_3 A_3 A_3 A_3 A_2 A_2 A_2 A_2$ $A_1 A_1 A_1 A_1$.
- -		$A_2 A_2 A_2 A_2$	$A_1 A_1 A_1 A_1$	
5 -	*JMS	0 1 0 1	$A_3 A_3 A_3 A_3$	Jump to subroutine ROM address $A_3 A_3 A_3 A_3 A_2 A_2 A_2 A_2$ $A_1 A_1 A_1 A_1$, save old address (up 1 level in stack.)
- -		$A_2 A_2 A_2 A_2$	$A_1 A_1 A_1 A_1$	
6 -	INC	0 1 1 0	R R R R	Increment contents of register RRRR.
7 -	*ISZ	0 1 1 1	R R R R	Increment contents of register RRRR. Go to ROM address $A_2 A_2 A_2 A_2$ $A_1 A_1 A_1 A_1$ (within the same ROM that contains this ISZ instruction) if result = 0, otherwise go to the next instruction in sequence.
- -		$A_2 A_2 A_2 A_2$	$A_1 A_1 A_1 A_1$	
8 -	ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with carry.
9 -	SUB	1 0 0 1	R R R R	Subtract contents of register RRRR to accumulator with borrow.
A -	LD	1 0 1 0	R R R R	Load contents of register RRRR to accumulator.
B -	XCH	1 0 1 1	R R R R	Exchange contents of index register RRRR and accumulator.
C -	BBL	1 1 0 0	D D D D	Branch back (down 1 level in stack) and load data DDDD to accumulator.
D -	LDM	1 1 0 1	D D D D	Load data DDDD to accumulator.
F0	CLB	1 1 1 1	0 0 0 0	Clear both. (Accumulator and carry)
F1	CLC	1 1 1 1	0 0 0 1	Clear carry.
F2	IAC	1 1 1 1	0 0 1 0	Increment accumulator.
F3	CMC	1 1 1 1	0 0 1 1	Complement carry.
F4	CMA	1 1 1 1	0 1 0 0	Complement accumulator.
F5	RAL	1 1 1 1	0 1 0 1	Rotate left. (Accumulator and carry)
F6	RAR	1 1 1 1	0 1 1 0	Rotate right. (Accumulator and carry)
F7	TCC	1 1 1 1	0 1 1 1	Transmit carry to accumulator and clear carry.
F8	DAC	1 1 1 1	1 0 0 0	Decrement accumulator.
F9	TCS	1 1 1 1	1 0 0 1	Transfer carry subtract and clear carry.
FA	STC	1 1 1 1	1 0 1 0	Set carry.
FB	DAA	1 1 1 1	1 0 1 1	Decimal adjust accumulator.
FC	KBP	1 1 1 1	1 1 0 0	Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.
FD	DCL	1 1 1 1	1 1 0 1	Designate command line.

4040 ONLY INSTRUCTIONS

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
01	HLT	0 0 0 0	0 0 0 1	Executes Halt until interrupt received.
02	BBS	0 0 0 0	0 0 1 0	Return from subroutine and restore SRC:
03	LCR	0 0 0 0	0 0 1 1	Data RAM and ROM bank status loaded into ACC.
04	OR4	0 0 0 0	0 1 0 0	OR accumulator with IR4.

4040 ONLY INSTRUCTIONS (Continued)

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
05	OR5	0 0 0 0	0 1 0 1	OR accumulator with IR5.
06	AN6	0 0 0 0	0 1 1 0	AND accumulator with IR6.
07	AN7	0 0 0 0	0 1 1 1	AND accumulator with IR7.
08	DBO	0 0 0 0	1 0 0 0	Select ROM bank 0.
09	DB1	0 0 0 0	1 0 0 1	Select ROM bank 1.
0A	SBO	0 0 0 0	1 0 1 0	Select IR bank 0.
0B	SB1	0 0 0 0	1 0 1 1	Select IR bank 1.
0C	EIN	0 0 0 0	1 1 0 0	Enable interrupt detection.
0D	DIN	0 0 0 0	1 1 0 1	Disable interrupt detection.
0E	RPM	0 0 0 0	1 1 1 0	Load accumulator from 4289-controlled program RAM.

4001/4002/4008/4009/4289 INPUT/OUTPUT AND RAM INSTRUCTIONS

Hex Code	MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
2 -	SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X_2 and X_3 time in the instruction cycle.
E0	WRM	1 1 1 0	0 0 0 0	Write the contents of the accumulator into the previously selected RAM main memory character.
E1	WMP	1 1 1 0	0 0 0 1	Write the contents of the accumulator into the previously selected RAM output port. (Output Lines)
E2	WRR	1 1 1 0	0 0 1 0	Write the contents of the accumulator into the previously selected ROM output port. (I/O Lines)
E3	WPM	1 1 1 0	0 0 1 1	Write the contents of the accumulator into the previously selected half byte of read/write program memory (used with 4008/4009 or 4289 only)
E4	WR0	1 1 1 0	0 1 0 0	Write the contents of the accumulator into the previously selected RAM status character 0.
E5	WR1	1 1 1 0	0 1 0 1	Write the contents of the accumulator into the previously selected RAM status character 1.
E6	WR2	1 1 1 0	0 1 1 0	Write the contents of the accumulator into the previously selected RAM status character 2.
E7	WR3	1 1 1 0	0 1 1 1	Write the contents of the accumulator into the previously selected RAM status character 3.
E8	SBM	1 1 1 0	1 0 0 0	Subtract the previously selected RAM main memory character from accumulator with borrow.
E9	RDM	1 1 1 0	1 0 0 1	Read the previously selected RAM main memory character into the accumulator.
EA	RDR	1 1 1 0	1 0 1 0	Read the contents of the previously selected ROM input port into the accumulator. (I/O Lines)
EB	ADM	1 1 1 0	1 0 1 1	Add the previously selected RAM main memory character to accumulator with carry.
EC	RD0	1 1 1 0	1 1 0 0	Read the previously selected RAM status character 0 into accumulator.
ED	RD1	1 1 1 0	1 1 0 1	Read the previously selected RAM status character 1 into accumulator.
EE	RD2	1 1 1 0	1 1 1 0	Read the previously selected RAM status character 2 into accumulator.
EF	RD3	1 1 1 0	1 1 1 1	Read the previously selected RAM status character 3 into accumulator.